

Расчетно-графическая работа

«Использование нейронных сетей для задач классификации и распознавания»

1. Теория

1.1 Нейронные сети

Искусственные нейронные сети (Artificial neural network) – математические модели, а также их программные или аппаратные реализации, построенные по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма.

На рисунке 1 приведена модель нейрона.

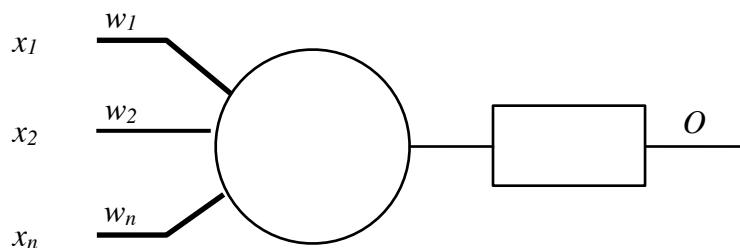


Рисунок 1. – Модель нейрона

На рисунке приняты следующие обозначения: x_i – сигнал на i -м входе (синапсе) нейрона; w_i – вес i -ого входа (синапса) нейрона; Z – взвешенная сумма входных воздействий на нейрон; $y = f(Z)$ – функция активации нейрона; O – выход нейрона. Примеры активационных функций будут рассмотрены ниже.

Из связанных определенным образом нейронов (узлов) строится нейронная сеть с некоторым количеством входных и выходных узлов. Нейронные сети различают по структуре сети, особенностям модели нейрона и подходам к обучению.

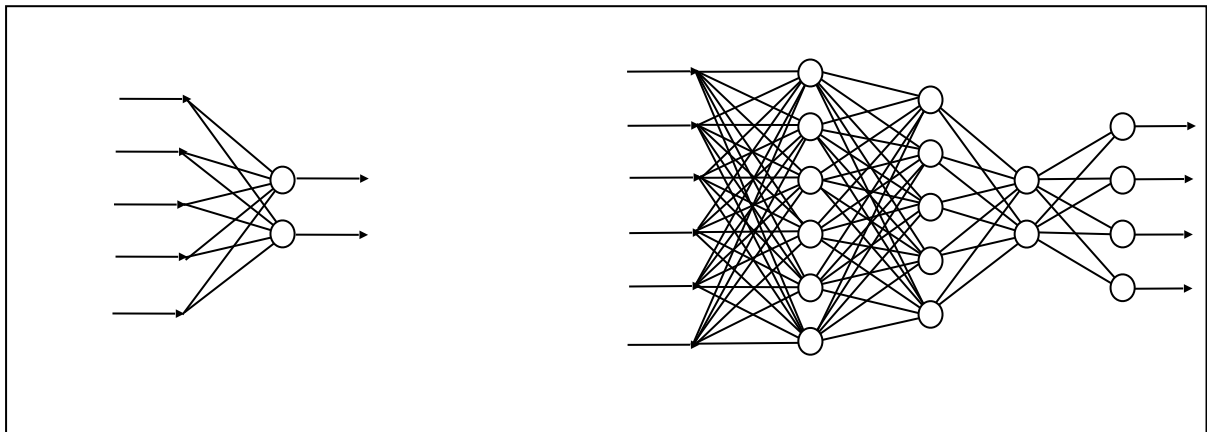


Рисунок 2. – Однослойная и многослойная сеть

По структуре нейронные сети можно разделить на неполносвязные и полносвязные. Неполносвязные сети (описываемые неполносвязным ориентированным графом и обычно называемые персептронами) в свою очередь делятся на однослойные и многослойные, многослойные сети делятся на сети с прямыми связями, с перекрестными связями и с обратными связями. В нейронных сетях с прямыми связями нейроны j -ого слоя по входам могут соединяться только с нейронами i -х слоев, где $j > i$, т.е. все нейроны разбиты на слои и связи возможны лишь между нейронами двух соседних слоев и лишь в одном направлении. Наиболее популярными являются многослойные сети с прямыми связями (сети прямого распространения) в связи с простотой их моделирования и обучения, на

рисунке 2 приведены примеры структуры однослойной и многослойной сети прямого распространения.

Кроме структуры сети, нейронные сети можно классифицировать по используемым подходам к обучению, по типу активационной функции используемой в нейронах и по другим признакам.

Активационные функции

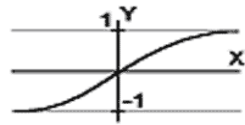
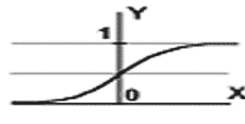
Первой активационной функцией предложенной для использования в искусственных нейронных сетях является функция единичного скачка:

$$f(x) = \begin{cases} 0, & x < h \\ 1, & x \geq h \end{cases}$$

Таким образом, в модели взвешенная сумма входных сигналов сравнивается с пороговым значением h , и на выходе появляется сигнал, если сумма превышает порог.

В современных нейронных сетях функция единичного скачка (пороговая функция) заменяется на нелинейную дифференцируемую активационную функцию, см. примеры функций в таблице 1.

Таблица 1. – Примеры нелинейных активационных функций

| Активационная функция | Формула | Вид |
|---|--|---|
| Гиперболический тангенс | $f(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}}$ |  |
| Логистическая (экспоненциальная сигмоида) | $f(x) = \frac{1}{1 + e^{-ax}}$ |  |
| Softmax функция | $f(x, i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$ | |

Обучение нейронных сетей

Нейронные сети можно классифицировать по типу обучения. Первый тип – обучение с учителем, суть которого заключается в следующем:

- Подготавливается набор обучающих данных, представляющих собой примеры входных данных и соответствующих им выходов;
- Нейронная сеть обучается с помощью того или иного алгоритма управляемого обучения, при котором имеющиеся данные используются для корректировки параметров сети таким образом, чтобы минимизировать ошибку выхода сети на обучающем множестве.

Второй подход – обучение без учителя, обучающие данные содержат лишь значения входных переменных, но не содержат примеров выходов.

Основным алгоритмом обучения с учителем в многослойных сетях прямого распространения является алгоритм обратного распространения.

Алгоритм обратного распространения – это итеративный градиентный алгоритм, который используется с целью минимизации ошибки (например, среднеквадратичного отклонения) текущего выхода многослойной сети и желаемого выхода. Основное

требование данного алгоритма к сети прямого распространения – дифференцируемость активационной функции.

Алгоритм состоит из следующих шагов:

1. Инициализировать (синаптические) веса маленькими случайными значениями.
2. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
3. Вычислить выход сети с помощью прямого распространения.
4. Вычислить разность между выходом сети и требуемым выходом, вычислить градиенты для выходного слоя.
5. С помощью обратного распространения ошибки вычислить градиенты для скрытых слоев
6. Подкорректировать веса сети для минимизации ошибки.
7. Повторять шаги с 2 по 5 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Одна из наиболее серьезных трудностей обучения с учителем заключается в том, что таким образом мы минимизируем не ту ошибку, которую на самом деле нужно минимизировать, - ошибку, которую можно ожидать от сети, когда ей будут подаваться совершенно новые наблюдения. Иначе говоря, мы хотели бы, чтобы нейронная сеть обладала способностью обобщать результат на новые наблюдения. В действительности же сеть обучается минимизировать ошибку на обучающем множестве, и в отсутствие идеального и бесконечно большого обучающего множества это совсем не то же самое, что минимизировать "настоящую" ошибку на поверхности ошибок в заранее неизвестной модели явления.

Сильнее всего **это различие проявляется в проблеме переобучения**, или слишком близкой подгонки. Пример с полиномами – полином низкого порядка может быть недостаточно гибким средством для аппроксимации данных, в то время как полином высокого порядка может оказаться чересчур гибким, будет проходить через все точки (следовать данным), при этом принимая сложную форму в интервалах между ними, не имеющую никакого отношения к форме настоящей зависимости.

2. Предобработка данных

2.1 Для массива данных MNIST:

1. Необходимо разрезать картинку с рукописными символами на отдельные символы. Размер отдельного символа 28x28 пикселей.
2. Выделить для каждого сегментированного изображения набор информативных признаков, которые будут использованы в качестве входных данных для нейронной сети. Методы выделения признаков (feature extraction) для изображения описаны ниже.

Для предобработки изображений можно воспользоваться библиотекой OpenCV для Python (http://opencv24-python-tutorials.readthedocs.io/en/stable/py_tutorials/py_tutorials.html)

Каждый символ имеет свои уникальные признаки (положение и наклон линий, дуг, наличие петель и др.), которые позволяют их различать. Аналогичным образом системы автоматизированного распознавания символов осуществляют принятие решений на основе некоторого описания объекта посредством набора (системы) признаков. Однако использование неинформативных и избыточных признаков не только оказывается бесполезным, но и снижает эффективность процесса распознавания. Поэтому выбор информативных признаков имеет большое значение при разработке систем распознавания образов и распознавания символов в частности.

На сегодняшний день предложено довольно много различных методов выделения информативных признаков для распознавания символов, но в рамках учебного процесса мы ограничимся рассмотрением лишь некоторых из них, а в частности:

1. метода зон;

2. метода проекционных гистограмм;
3. метода проекций.

Метод зон

Область, содержащая символ, разбивается на несколько перекрывающихся или не перекрывающихся зон, после чего для каждой зоны вычисляется число пикселей, принадлежащих распознаваемому объекту, как правило, нормализованное на общее число пикселей в зоне. Образно этот процесс можно представить с помощью рисунка 3. Чем темнее квадрат, тем соответственно выше плотность пикселей, принадлежащих распознаваемому символу.

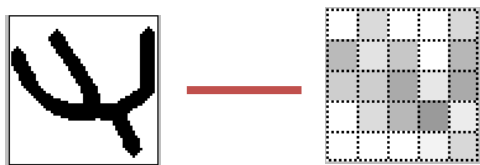


Рисунок 3. – Иллюстрация выделения информативных признаков методом зон

Метод возвращает вектор признаков (f_1, \dots, f_k) , где f_i – нормализованное число черных пикселей в i -ой зоне.

Метод проекционных гистограмм

Метод проекционных гистограмм вычисляет количество пикселей, принадлежащих распознаваемому символу, в заданном направлении. Выделяют три типа проекционных гистограмм: горизонтальные, вертикальные и диагональные.

На рисунке 4 представлены вертикальная и горизонтальная проекционные гистограммы для символа 'а'.

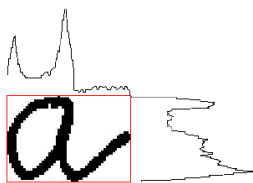


Рисунок 4. – Вертикальная и горизонтальная проекционные гистограммы для символа 'а'

В случае совместного использования горизонтальных и вертикальных проекционных гистограмм, метод возвращает вектор признаков $(f_{v1}, \dots, f_{vm}, f_{h1}, \dots, f_{hn})$, где f_{vi} – количество черных пикселей в i -ой строке, а f_{hj} – количество черных пикселей в j -ом столбце.

Метод проекций

Область размером $l \times m$, содержащая символ, разбивается на k областей, как показано на рисунке 5, и затем для каждого региона строится проекция части распознаваемого объекта.

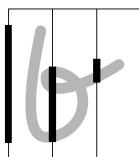


Рисунок 5. – Иллюстрация метода проекций

В случае горизонтальных проекций, метод возвращает вектор признаков (f_1, \dots, f_k) , где $f_r = (p_1, \dots, p_m)$, $p_i = I(i, j)$ – функция принадлежности к распознаваемому объекту пикселя в i -ой строке и j -ом столбце.

Метод можно расширить на случай вертикальных проекций и проекций по произвольным направлениям.

2.2 Для массивов данных по классификации:

Не для всех массивов данных по классификации нужна предобработка данных, но для некоторых могут понадобиться следующие процедуры:

1. Восстановление пропущенных значений
2. Преобразование данных из номинальных шкал в численные атрибуты (например, данные об образовании). Для этой задачи можно воспользоваться классом `OneHotEncoder` библиотеки `Scikit-learn` (<http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html#sklearn.preprocessing.OneHotEncoder>).
3. Нормировка данных, если значения разных атрибутов различаются на порядки. Это позволит улучшить качество, получаемых моделей нейронных сетей. Для этой задачи можно использовать функцию `scale` библиотеки `Scikit-learn` (<http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.scale.html#sklearn.preprocessing.scale>).

3. Исходные данные

Массив рукописных изображений MNIST

Массив уже заранее поделен на обучающую и тестовую выборку. Все изображения в рамках одной цифры склеены в одну картинку. Размер области одной картинки составляет 28x28 пикселей. Все картинки уже переведены в монохромные изображения.

Массивы данных для классификации

Массивы представляют данные различных предметных областей (медицина, сельское хозяйство, описание социальных групп и т.д.), используемые в задачах классификации. Все массивы взяты с открытого репозитория данных UC Irvine Machine Learning Repository (<http://archive.ics.uci.edu/ml/index.html>). Для каждого варианта имеется: html-файл, описывающий структуру исходного массива данных, и текстовый файл, содержащий собственно сам массив данных.

4. Инструменты

Нейронные сети в Scikit-learn

Для работы с нейронами сетями используется реализация многослойного перестроена `MLPClassifier`. Ссылки на официальную документацию:

- http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier
- http://scikit-learn.org/stable/modules/neural_networks_supervised.html#tips-on-practical-use

Пример обучения нейронной сети:

```
>>> from sklearn.neural_network import MLPClassifier
>>> X = [[0., 0.], [1., 1.]]
>>> y = [0, 1]
>>> clf = MLPClassifier(solver='lbfgs', activation='relu',
                        hidden_layer_sizes=(5, 2), random_state=1)
>>> clf.fit(X, y)
>>> clf.predict([[2., 2.], [-1., -2.]])
```

Особое внимание стоит обратить на два параметра конструктора `MLPClassifier`, используемые в работе: `activation` используется для того, чтобы задать вид активационной функции, `hidden_layer_sizes` используется для задания внутренних слоев нейронной сети. Количество нейронов в скрытых слоях подбирается опытным. Тем не менее, существует эвристическое правило, согласно которому число нейронов в скрытом слое - это корень квадратный от произведения числа нейронов во входном слое на число нейронов в

выходном слое. Количество нейронов во входном и выходном слоях автоматически определяется согласно размерности исходных данных.

Для разделения массива на обучающую и тестовую выборку (массивы для классификации) можно использовать ***train_test_split*** - метод для разделения произвольного массива данных на обучающую и тестовую выборку. Массив данных при этом перемешивается.

Пример использования:

```
from sklearn.cross_validation import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(train_data_features,
Y, test_size=0.2, random_state=0)
```

В параметре ***test_size*** указывается относительная доля тестовой выборки в исходном массиве данных.

Для оценки точности можно использовать реализованную метрику ***accuracy_score*** - метод для оценки качества классификации.

Пример использования:

```
from sklearn import metrics
print metrics.accuracy_score(Y_test, res_clf)
```

Y_test - реальные значения тестовой выборки, ***res_clf*** - результат, полученный с помощью классификатора.

5. Задание

1. Изучить структуру исходных данных. Если вы используете один из массивов для классификации, то по структуре исходных данных необходимо определить выходную переменную.
2. Произвести предобработку исходных данных. Процедура предобработки зависит от используемого массива данных и описана в пункте 2 «Предобработка данных».
3. Согласно своему варианту спланировать схему экспериментов по обучению и тестированию нейронной сети. Например, исходя из количества слоев, функции активации и соотношения обучающей и тестовой выборок.
4. Используя реализацию многослойного перестроена библиотеки Scikit-learn реализовать сетку экспериментов. Для расчета качества полученных моделей нейронных сетей использовать метрику точности (доля правильных ответов нейронной сети на тестовой выборке).
5. Сделать выводы по качеству полученных моделей НС.

2. Варианты

Задача классификации

В каждой задаче используется разный массив исходных данных. Для каждого варианта есть его описание (html-файл) и сам массив данных в текстовом файле.

Согласно описанию массива исходных данных необходимо выделить выходную переменную, которая будет использована для прогнозирования (ответа нейронной сети).

| Вариант | Количество слоев | Функции активации (параметр activation) | Обучающая/тестовая выборка |
|------------|------------------|---|----------------------------|
| 1 (усложн) | 4-7 | identity, logistic | 70/30, 90/10 |
| 2 | 2-5 | tanh, relu | 60/40, 80/20 |
| 3 | 3-6 | identity, relu | 80/20, 90/10 |
| 4 | 4-7 | tanh, logistic | 70/30, 90/10 |
| 5 | 2-5 | identity, logistic | 60/40, 80/20 |
| 6 | 3-6 | tanh, relu | 80/20, 90/10 |

| | | | |
|------------|-----|--------------------|--------------|
| 7 | 2-5 | identity, relu | 70/30, 90/10 |
| 8 (усложн) | 4-7 | tanh, logistic | 80/20, 90/10 |
| 9 | 2-5 | identity, logistic | 70/30, 90/10 |
| 10 | 3-6 | tanh, relu | 60/40, 80/20 |
| 11 | 4-7 | identity, relu | 80/20, 90/10 |

Задача распознавания

Используется массив данных MNIST. Для каждого из вариантов создается подвыборка (только для распознаваемых цифр) согласно варианту. Массив данных уже разделен на обучающую и тестовую выборку (папки train и test).

| Вариант | Распознаваемые цифры | Количество слоев | Функции активации (параметр activation) | Метод предобработки |
|---------|----------------------|------------------|---|---|
| 12 | 1,3,5 | 4-7 | identity, logistic | метод зон, метод проекций |
| 13 | 2,4,6 | 2-5 | tanh, relu | метод зон, метод проекционных гистограмм |
| 14 | 3,5,7 | 3-6 | identity, relu | метод проекций, метод проекционных гистограмм |
| 15 | 4,6,8 | 4-7 | tanh, logistic | метод зон, метод проекций |
| 16 | 5,7,9 | 2-5 | identity, logistic | метод зон, метод проекционных гистограмм |
| 17 | 6,8,0 | 3-6 | tanh, relu | метод проекций, метод проекционных гистограмм |
| 18 | 7,9,1 | 2-5 | identity, relu | метод зон, метод проекций |
| 19 | 8,0,2 | 4-7 | tanh, logistic | метод зон, метод проекционных гистограмм |
| 20 | 1,3,9 | 2-5 | identity, logistic | метод проекций, метод проекционных гистограмм |
| 21 | 0,2,4 | 3-6 | tanh, relu | метод зон, метод проекций |
| 22 | 1,4,7 | 4-7 | identity, relu | метод зон, метод проекционных гистограмм |
| 23 | 2,3,4 | 4-7 | identity, logistic | метод проекций, метод проекционных гистограмм |

| | | | | |
|----|-------|-----|----------------|--|
| 24 | 4,7,8 | 2-5 | tanh, relu | метод зон, метод проекций |
| 25 | 0,2,7 | 3-6 | identity, relu | метод зон, метод проекционных гистограмм |