

Искусственные нейронные сети

Практические задания для самостоятельного выполнения

Задание 11.

Создать и обучить нейросеть, решающую задачу аппроксимации функции пяти независимых переменных. В качестве аппроксимируемой функции рассмотреть

$$y = \frac{x_1 + x_2}{x_3} + x_4 \cdot x_5.$$

1. Сформировать набор данных из 1000 объектов: значения признаков – псевдослучайные числа, равномерно распределенные на интервале (1; 3); значения функции определяются по указанной выше формуле. Обеспечить воспроизводимость результатов.
Внимание: значения признаков должны быть независимыми!
2. Выполнить разбиение данных на обучающую/тестовую выборки (обучающая выборка – 70% от всех имеющихся данных). Сформировать обучающий и тестовый набор для нейросети.
3. Создать нейросеть с одним скрытым слоем, состоящим из двух нейронов. Обучить ее на обучающем наборе данных. Максимальное количество эпох задать равным 50. Вывести аппроксимируемые значения функции и предсказания сети для тестовой выборки, а также ошибку сети на тестовой выборке.
4. Вывести график ошибок сети на обучающей и тестовой выборке в зависимости от количества итераций. Проанализировать полученный график и ответить на вопрос: как (предположительно) изменится ошибка сети при увеличении числа эпох до 100, 500, ...
5. Оценить рекомендуемое количество скрытых нейронов в задаче рассматриваемого класса. Создать новую сеть с таким количеством скрытых нейронов (количество скрытых слоев – 1). Обучить эту сеть на том же наборе данных, что и сеть из п. 3. Сравнить качество полученных аппроксимаций. Сделать выводы.
6. Внести изменения (шум) в исходный набор данных. Для этого прибавить к аппроксимируемым значениям функции нормально распределенную случайную величину со средним значением, равным нулю, и средним квадратичным отклонением σ , равным 0.2.
7. Выполнить обучение нейросети на зашумленных данных (использовать сеть с той же структурой и те же параметры обучения, что и для исходного набора данных). Сравнить полученные результаты.
8. Поэкспериментировать со степенью зашумленности исходных данных, варьируя значения σ . Проанализировать зависимость качества аппроксимации от значения σ .

Задание 12.

Создать и обучить нейросеть, аппроксимирующую таблицу умножения.

1. Сформировать набор данных: таблица умножения для чисел от 1 до 10. Выполнить разбиение данных на обучающую/тестовую выборки (обучающая выборка – 70% от всех имеющихся данных). Сформировать обучающий и тестовый набор для нейросети.
2. Создать нейросеть с одним скрытым слоем, состоящим из двух нейронов. Обучить ее на обучающем наборе данных. Вывести аппроксимируемые значения и предсказания сети для тестовой выборки. Вывести график ошибок сети на обучающей и тестовой выборке в зависимости от количества итераций. Сделать выводы.
3. Поэкспериментировать с числом скрытых нейронов и числом скрытых слоев (число слоев не более 2). Проанализировать полученные результаты, сделать выводы.
4. Выполнить масштабирование набора данных и повторить действия п. 2. Сравнить полученные результаты с результатами выполнения п. 2.
5. Поэкспериментировать с параметрами алгоритма обучения: длиной шага и уменьшением длины шага (в классе *BackpropTrainer* – *learningrate* и *lrdecay* соответственно), а также с количеством эпох. Определить набор значений параметров, обеспечивающий наилучшие результаты.

Задание 13 (дополнительно).

Рассматривается задача оценки качества вина по его физико-химическим свойствам: P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. *Modeling wine preferences by data mining from physicochemical properties*. In *Decision Support Systems*, Elsevier, 47(4):547-553, 2009.

Исходные данные размещены в открытом доступе: <https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv> (репозиторий UCI). Они содержат 1599 образцов красного вина, описанных 11 признаками, среди которых – кислотность, процентное содержание сахара, алкоголя и др. Кроме того, каждому объекту поставлена в соответствие оценка качества по шкале от 0 до 10 (выставлена экспертами). Требуется восстановить оценку качества вина по исходному признаковому описанию.

1. Проанализировать задачу и формализовать ее как задачу многоклассовой классификации.
2. Импортировать исходные данные в массив *NumPy* (проще всего – использовать метод *numpy.loadtxt*). Вывести несколько первых записей и размерность сформированного массива (для контроля корректности импорта). Указание. Альтернативой загрузки данных из заранее сохраненного файла может служить загрузка данных напрямую из репозитория. Для этого следует использовать метод *urlopen* стандартного модуля *urllib.request*.
3. Выделить целевой признак (экспертную оценку) в отдельный массив. Вывести содержимое этого массива в виде гистограммы и убедиться, что данные являются несбалансированными: основная масса объектов имеет оценки от 5 до 7.
4. Привести рассматриваемую задачу к задаче трехклассовой классификации: объектам с оценкой менее 5 сопоставить оценку 5, объектам с оценкой более 7

- оценку 7; после этого – вычесть из всех полученных оценок минимальную. В итоге должен получиться набор данных с оценками 0, 1 и 2.
5. Выполнить стандартную нормализацию признаков объектов (написав простой код либо применив метод *normalize* модуля *sklearn.preprocessing*).
 6. Выполнить разбиение данных на обучающую/тестовую выборки (обучающая выборка – 70% от всех имеющихся данных).
 7. Инициализировать структуру данных *ClassificationDataSet*, используемую библиотекой *Pybrain* (структура принимает в качестве аргументов количество признаков в наборе данных и количество различных меток классов). Инициализацию следует выполнить отдельно для обучающего и тестового набора данных (создать два объекта). В каждом из созданных объектов выполнить бинаризацию целевой переменной с помощью функции *_convertToOneOfMany()*.
 8. Создать нейросеть с одним скрытым слоем, состоящим из 100 нейронов; функция активации выходного слоя – *softmax*. Обучить сеть на обучающем наборе данных, задав максимальное количество эпох равным 100. Вывести график ошибок сети на обучающей и тестовой выборке в зависимости от количества итераций. Вывести доли неправильных ответов на обучающей и тестовой выборке.
 9. Проанализировать полученные результаты. Сформулировать свои предложения по улучшению качества модели. Выполнить их реализацию.