Обучение на ошибках

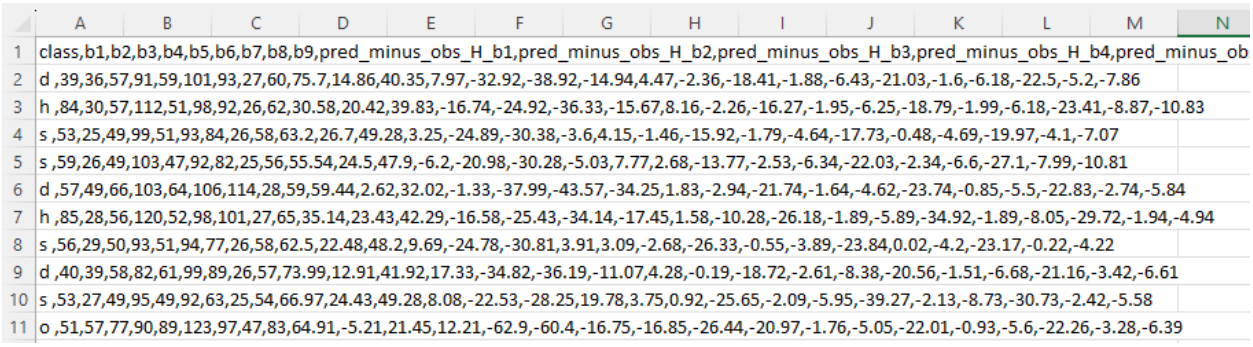
Процесс обучения нейронной сети с использованием дельта-правила с редукцией – это процесс детерминированный и гарантированно сходящийся.

Например, для обучающей выборки набора данных Forest type mapping Data Set можно получить нейронную сеть, которая не имеет ошибок на тестовой и обучающей выборках путем переноса ошибочных экземпляров из тестовой выборки в обучающую.

Датасет находится по следующей ссылке:

https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping

Он представлен в веществennозначном виде:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	class,b1,b2,b3,b4,b5,b6,b7,b8,b9,pred_minus_obs_H_b1,pred_minus_obs_H_b2,pred_minus_obs_H_b3,pred_minus_obs_H_b4,pred_minus_ob													
2	d ,39,36,57,91,59,101,93,27,60,75.7,14.86,40.35,7.97,-32.92,-38.92,-14.94,4.47,-2.36,-18.41,-1.88,-6.43,-21.03,-1.6,-6.18,-22.5,-5.2,-7.86													
3	h ,84,30,57,112,51,98,92,26,62,30.58,20.42,39.83,-16.74,-24.92,-36.33,-15.67,8.16,-2.26,-16.27,-1.95,-6.25,-18.79,-1.99,-6.18,-23.41,-8.87,-10.83													
4	s ,53,25,49,99,51,93,84,26,58,63.2,26.7,49.28,3.25,-24.89,-30.38,-3.6,4.15,-1.46,-15.92,-1.79,-4.64,-17.73,-0.48,-4.69,-19.97,-4.1,-7.07													
5	s ,59,26,49,103,47,92,82,25,56,55.54,24.5,47.9,-6.2,-20.98,-30.28,-5.03,7.77,2.68,-13.77,-2.53,-6.34,-22.03,-2.34,-6.6,-27.1,-7.99,-10.81													
6	d ,57,49,66,103,64,106,114,28,59,59.44,2.62,32.02,-1.33,-37.99,-43.57,-34.25,1.83,-2.94,-21.74,-1.64,-4.62,-23.74,-0.85,-5.5,-22.83,-2.74,-5.84													
7	h ,85,28,56,120,52,98,101,27,65,35.14,23.43,42.29,-16.58,-25.43,-34.14,-17.45,1.58,-10.28,-26.18,-1.89,-5.89,-34.92,-1.89,-8.05,-29.72,-1.94,-4.94													
8	s ,56,29,50,93,51,94,77,26,58,62.5,22.48,48.2,9.69,-24.78,-30.81,3.91,3.09,-2.68,-26.33,-0.55,-3.89,-23.84,0.02,-4.2,-23.17,-0.22,-4.22													
9	d ,40,39,58,82,61,99,89,26,57,73.99,12.91,41.92,17.33,-34.82,-36.19,-11.07,4.28,-0.19,-18.72,-2.61,-8.38,-20.56,-1.51,-6.68,-21.16,-3.42,-6.61													
10	s ,53,27,49,95,49,92,63,25,54,66.97,24.43,49.28,8.08,-22.53,-28.25,19.78,3.75,0.92,-25.65,-2.09,-5.95,-39.27,-2.13,-8.73,-30.73,-2.42,-5.58													
11	o ,51,57,77,90,89,123,97,47,83,64.91,-5.21,21.45,12.21,-62.9,-60.4,-16.75,-16.85,-26.44,-20.97,-1.76,-5.05,-22.01,-0.93,-5.6,-22.26,-3.28,-6.39													

Варианты обучения нейронной сети wide_train.ipynb содержатся в папках «1_first_train», «2_add_12_92», «3_add_10», «4_add_125» репозитория Github.

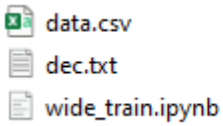
Создадим тестовую папку test, в которой будут загружены соответствующие файлы обучений на ошибках.

1) Первое обучение

Проведем первое обучение нейронной сети. Для этого в каталоге test создадим папку 1_first_train и поместим туда скачанный набор данных data.csv из папки «1_first_train» репозитория Github. Дальнейшие действия для первого обучения будут проведены в этой папке. Создадим файл «dec.txt», который содержит количество десятичных знаков после запятой

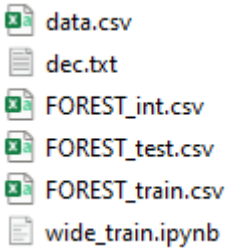
для каждого из столбцов набора данных (в нашем случае этот файл содержит 27 строк). Загрузим в созданную нами папку 1_first_train также соответствующий файл wide_train.ipynb из папки «1_first_train» репозитория Github.

Таким образом, на локальном компьютере создана тестовая папка test, в которой добавлена папка 1_first_train, соответствующая первому обучению нейронной сети, со следующими файлами:



В файле wide_train.ipynb содержатся команды для проведения обучения. Сначала происходит импорт библиотеки widelearning, далее обучение с выбором целевого класса, и в конце проверка результатов обучения.

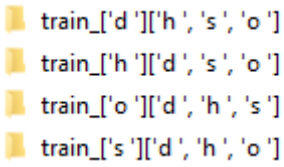
Для первого обучения необходимо преобразовать набор данных data.csv в целочисленный вид с использованием функции data_int. В результате выполнения преобразования папка 1_first_train выглядит следующим образом:



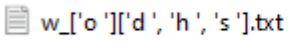
Среди всех файлов csv в папке 1_first_train необходимо оставить FOREST_train.csv и FOREST_test.csv. Остальные csv файлы можно удалить. Файл FOREST_train.csv, который является обучающей выборкой для первого нейрона, необходимо переименовать в train_1.csv. Создадим в папке 1_first_train две папки: data и weights. В папку data будут помещены обучающие выборки для каждого нейрона, в папку weights – веса каждого

нейрона. До проведения обучения обучающая выборка для первого нейрона train_1.csv должна быть помещена в папку data.

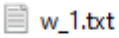
Начнем обучение первого нейрона, указав путь к соответствующей обучающей выборке (train_1.csv). В результате выполнения функции select_top сгенерировались следующие папки, в которых значение в первых квадратных скобках обозначает целевой класс:



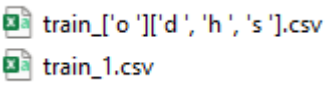
Целевой класс выбирается по максимальной доле отсеченных экземпляров «сверху». В данном случае на первом нейроне целевым классом выбирается класс «o». Открываем соответствующую папку и копируем файлы w.txt и train.csv в папки data и weights, переименовав их после перемещения. То есть выбираем соответствующий файл весов для выбранного целевого класса:



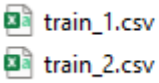
Копируем этот файл весов в папку weights и переименовываем:



Выбираем соответствующую обучающую выборку для следующего нейрона и копируем ее в папку data каталога 1_first_train:

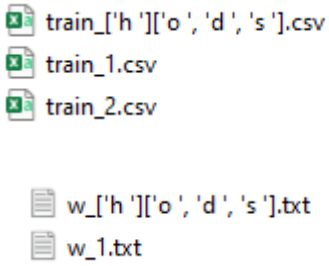


Далее переименовываем эту обучающую выборку:

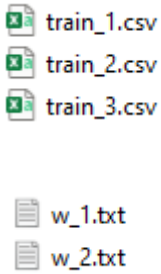


Удаляем папки, сгенерированные после select_top.

Обучаем второй нейрон, выбираем соответствующую целевому классу папку (класс «h»), копируем оттуда обучающую выборку для следующего нейрона и файл весов текущего нейрона в папки data и weights:

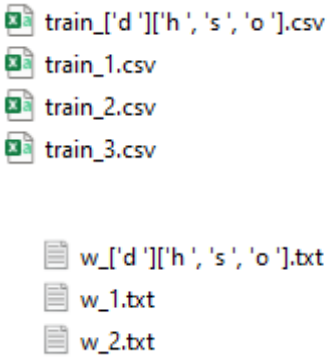


Переименовываем следующим образом:

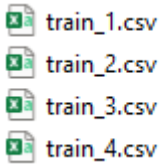


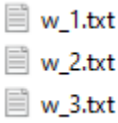
Удаляем сгенерированные папки.

Обучаем третий нейрон. Копируем обучающую выборку и файл весов в папки data и weights (класс «d»).



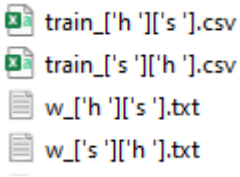
Переименовываем файлы:



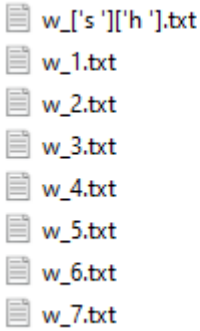


Дальнейшее обучение выполняет аналогичным образом.

На восьмом нейроне в обучающей выборке остаются два класса: «h» и «s». Для обучения используется вариант функции для бинарного датасета с пометкой binary. В данном случае файлы train.csv и w.txt генерируются прямо в папку 1_first_train, без вложения в соответствующие папки:



На восьмом (последнем) нейроне целевым выбирается класс «s». Копируется соответствующий файл весов в папку weights:

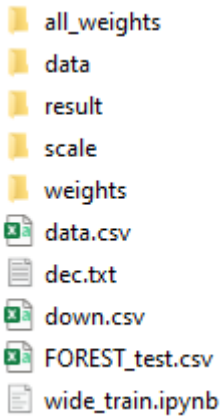


Далее этот файл весов переименовывается как w_8.txt.

Обучающую выборку для следующего нейрона не нужно копировать в папку data, поскольку восьмой нейрон является последним. Обучающая выборка train_[‘s’][‘h’].csv является пустой, так как были разделены все классы.

Таким образом, после первого обучения папка data содержит восемь файлов train_1.csv…train_8.csv, папка weights содержит восемь файлов w_1.txt…w_8.txt.

Далее масштабируются полученные веса с помощью функции scale_weights. Генерируются дополнительные папки all_weights, scale, result:



Папки result и scale являются побочными для генерации файла txt со всеми весами, находящегося в папке all_weights.

Проведем проверку результатов обучения с помощью функции check_test. В результате первого обучения имеется нейронная сеть из восьми нейронов, содержащая две ошибки на тестовой выборке (экземпляры № 12 и 92).

2) Второе обучение

В каталоге test необходимо создать папку «2_add_12_92». В нее необходимо загрузить соответствующий файл с командами для обучения wide_train.ipynb из репозитория Github. Также необходимо скопировать тестовую выборку FOREST_test.csv. Далее создать папки data и weights. В папку data поместить обучающую выборку для первого нейрона train_1.csv, отличием которой от выборки для первого обучения является добавление ошибочных экземпляров тестовой выборки (№12 и №92) в конец выборки.

Обучение производится аналогичным первому обучению образом. В данном случае не используется в начале функции data_int, поскольку датасет уже преобразован к целочисленному виду.

На втором обучении количество нейронов сократилось до семи, на тестовой выборке имеется одна ошибка (экземпляр № 10).

3) Третье обучение

В каталоге test необходимо создать папку «3_add_10». Переместить в нее соответствующий файл wide_train.ipynb из репозитория Github, создать папки data и weights, скопировать тестовую выборку.

Соответствующий ошибочный экземпляр тестовой выборки переносится в обучающую выборку. Таким образом, обучающая выборка для третьего обучения содержит три дополнительных экземпляра: №12, №92, №10.

На третьем обучении количество нейронов вновь возросло до восьми, имеется одна ошибка на тестовой выборке (экземпляр №125).

4) Четвертое обучение

Необходимо в каталоге test создать папку «4_add_125». Скопировать в нее соответствующий файл wide_train.ipynb из репозитория Github, скопировать тестовую выборку FOREST_test.csv, создать папки data и weights. В папку data поместить обучающую выборку, которая содержит 4 ошибочных экземпляра из тестовой выборки: №12, №92, №10, №125.

В результате четвертого, последнего обучения структура нейронной сети содержит семь нейронов, на тестовой и обучающей выборках ошибок не содержится.

На других датасетах на перенос ошибочных экземпляров из тестовой в обучающую выборку может потребоваться большее количество времени. Очевидно, что в предельном случае, переместив все экземпляры тестовой выборки в обучающую, можно гарантировать абсолютную безошибочность обучения. Однако обучая нейронную сеть на полной выборке, во-первых, можно попасть в ситуацию переобучения, а во-вторых, на действительно больших датасетах из категории Big Data, повторяющиеся переобучения могут оказаться чрезмерно затратными.

Предлагается следующий вариант использования открытой библиотеки wideLearning. Исходная выборка делится на три части: обучающую, тестовую и корректирующую, процентное отношение экземпляров каждого класса во всех выборках должно совпадать. Обучающая выборка, по размеру, может быть значительно меньше корректирующей. Дельта-правило с редукцией обучает нейроны последовательно, друг за другом. В процесс обучения каждого нейрона надо добавить процедуру тестирования корректирующей выборки, в ней могут обнаружиться ошибочные экземпляры. Для нейронов с троичной пороговой функцией активации ошибкой является появление экземпляра нецелевого класса «правее» правой границы и/или появление экземпляра не противоположного класса «левее» левой границы разделяющего коридора. Следовательно, в обучающую выборку надо переносить не все ошибочные экземпляры корректирующей выборки, а всего два – самый «правый» и самый «левый». Далее происходит переобучение текущего нейрона. Если в корректирующей выборке больше ошибок нет, то производится усечение и обучающей, и корректирующей выборки. Таким образом обучающая выборка пополняется считанным количество экземпляров (пропорционально количеству нейронов), а подавляющая часть корректирующей выборки в обучении не участвует совсем, качество обучения по-прежнему оценивается по тестовой выборке. На объемных наборах данных это приведет к колоссальной экономии вычислительных ресурсов при гарантированной обучаемости нейронных сетей.