

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 3 по курсу Методы поддержки принятия решений

«Решение задач с использованием искусственной нейронной сети»

Предметная область: определение нарисованной цифры

14

(количество листов)

ИСПОЛНИТЕЛЬ:

студенты группы ИУ5-74

Повираева Марина
(Poviraeva@me.com)

Волобуев Василий

_____ 2017 г.
" __ " _____

ПРЕПОДАВАТЕЛЬ:

Терехов В.И.,
к.т.н., доцент

Кафедра ИУ-5.

Москва - 2017

Оглавление

Оглавление.....	2
Цель работы.....	3
Задание.....	3
Описание предметной области и выбранной задачи.....	4
Структура, основные параметры выбранной ИНС и блок-схема алгоритма обучения.....	6
ИНС обратного распространения ошибки.....	6
Описание программы, ее ключевые особенности и новшества.....	9
Протоколы проведенных экспериментов, с указанием погрешности распознавания.....	10
Однослойная нейронная сеть.....	10
Двуслойная нейронная сеть обратного распространения ошибки.....	13
Выводы.....	13
Используемая литература.....	14

Цель работы

Целью лабораторной работы является углубление и закрепление теоретических знаний, полученных на лекциях, приобретение практических навыков самостоятельного исследования при решении задач выбора, обучения и работы ИНС.

В процессе выполнения лабораторной работы по теме «Решение задач с использованием искусственной нейронной сети» студенты решают следующие задачи (задания):

- описывают предметную область и выбирают решаемую задачу (предпочтение должно отдаваться задачам практической направленности);
- определяют множество обучающих примеров;
- в зависимости от решаемой задачи выбирают структуру ИНС;
- выбирают алгоритм обучения ИНС;
- проводят обучение ИНС на тестовом множестве примеров с помощью выбранного алгоритма обучения;
- исследуют работу обученной ИНС в режиме распознавания.

Задание

Разработать (или использовать готовую) программу, которая обучает ИНС распознавать черно-белое (bitmap) изображение, состоящее не менее чем из 35 пикселей (матрица 10x14). При этом, ИНС должна иметь входы, ассоциированные с пикселями матрицы, и выход(ы), количество которых соответствует решаемой задаче и выбранной архитектуре.

В написанной или выбранной программе должна быть реализована возможность задания множества обучающих примеров в виде образов (n матриц размерностью 10x14), а также изменения величины коэффициента скорости обучения. Программа должна предусматривать два режима работы: обучения и распознавания. Обучение должно производиться с использованием алгоритма, соответствующего архитектуре выбранной для решения задачи ИНС. Вероятность распознавания обученной ИНС должна быть не менее 65%.

Описание предметной области и выбранной задачи

Была выбрана задача определения чёрно-белого числа размером 14x10.

Исходные изображения рукописных цифр с базы данных MNIST, размером 28x28, были предварительно сжаты до размеров 14x10, переведены в чёрно-белый режим — использовались в дальнейшем в качестве тестовых и тренировочных выборок.

Непосредственно были запрограммированы однослойная нейронная сеть с различными вариантами активационных функций и разными скоростями обучения, а также реализованная двуслойная нейронная сеть на основе сигма-функции и опробованы динамическая и статическая вариации скорости обучения.

Базисом для данной задачи является искусственный нейрон.

Каждый искусственный нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Искусственный нейрон обладает группой синапсов — однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон — выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Общий вид искусственного нейрона приведен на рис. 1.

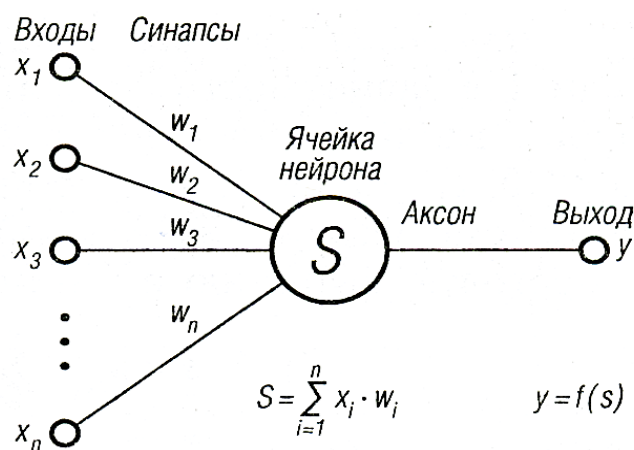


Рисунок 1

Искусственный нейрон в первом приближении имитирует свойства биологического нейрона. Здесь множество входных сигналов, обозначенных x_1, x_2, \dots, x_n , поступает на искусственный нейрон. Эти входные сигналы,

в совокупности обозначаемые вектором X , соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый синапс характеризуется величиной синаптической связи или ее весом w_i . Каждый сигнал умножается на соответствующий вес w_1, w_2, \dots, w_n , и поступает на суммирующий блок. Каждый вес соответствует «силе» одной биологической синаптической связи. (Множество весов в совокупности обозначаются вектором W) Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически, создавая величину S .

Таким образом, текущее состояние нейрона определяется как взвешенная сумма его входов:

$$S = \sum_{i=1}^n x_i \cdot w_i.$$

Выход нейрона есть функция его состояния: $y = f[S]$, где f — активационная функция, более точно моделирующая нелинейную передаточную характеристику биологического нейрона и предоставляющая нейронной сети большие возможности.

Пороговая функция ограничивает активность нейрона значениями 0 или 1 в зависимости от величины комбинированного входа s . Как правило, входные значения в этом случае также используются бинарные: $x_i \in [0, 1]$. Чаще всего удобнее вычесть пороговое значение θ , называемое смещением, из величины комбинированного входа и рассмотреть пороговую функцию в математически эквивалентной форме:

$$s = w_0 + \sum_{i=1}^n x_i \cdot w_i, f(s) = \begin{cases} 0, & s < 0 \\ 1, & s \geq 0 \end{cases}.$$

Здесь $w_0 = -\theta$ — величина смещения, взятая с противоположным знаком. Смещение обычно интерпретируется как связь, исходящая от элемента, значение которого всегда равно 1. Комбинированный вход тогда можно представить в виде $s = \sum_{i=0}^n x_i \cdot w_i$, где x_0 всегда считается равным 1.

Логистическая функция, или сигмоид, $f(s) = \frac{1}{1 + e^{-a \cdot s}}$ непрерывно заполняет своими значениями диапазон от 0 до 1; параметр a всегда положителен. При уменьшении параметра a график сигмоида становится более пологим, в пределе при $a = 0$ вырождаясь в горизонтальную линию на уровне 0,5, при увеличении параметра a график сигмоида приближается к виду функции единичного скачка с порогом 0. При этом, сигмоидальная функция дифференцируема на всей оси абсцисс, что используется

в некоторых алгоритмах обучения. Кроме того, она обладает свойством усиливать слабые сигналы и предотвращает насыщение от больших сигналов, так как они соответствуют тем областям аргументов, где сигмоид имеет пологий наклон

Структура, основные параметры выбранной ИНС и блок-схема алгоритма обучения

ИНС обратного распространения ошибки

Алгоритм обучения однослойного персептрона прост, однако много лет не удавалось обобщить этот алгоритм на случай многослойных сетей и только в 1986 году Румельхарт (D. E. Rumelhart) разработал эффективный алгоритм корректировки весов, названный алгоритмом обратного распространения ошибок (back propagation).

ИНС обратного распространения — это современный инструмент поиска закономерностей, прогнозирования и анализа. В них используется такой алгоритм обучения, в котором ошибка распространяется от выходного слоя к входному, т. е. в направлении, противоположном направлению распространения сигнала при нормальном функционировании сети.

Такая сеть состоит из нескольких слоев нейронов, причем каждый нейрон предыдущего слоя связан с каждым нейроном последующего слоя. В большинстве практических приложений оказывается достаточно рассмотрения двухслойной нейронной сети, имеющей входной (скрытый) слой нейронов и выходной слой (рис. 2).

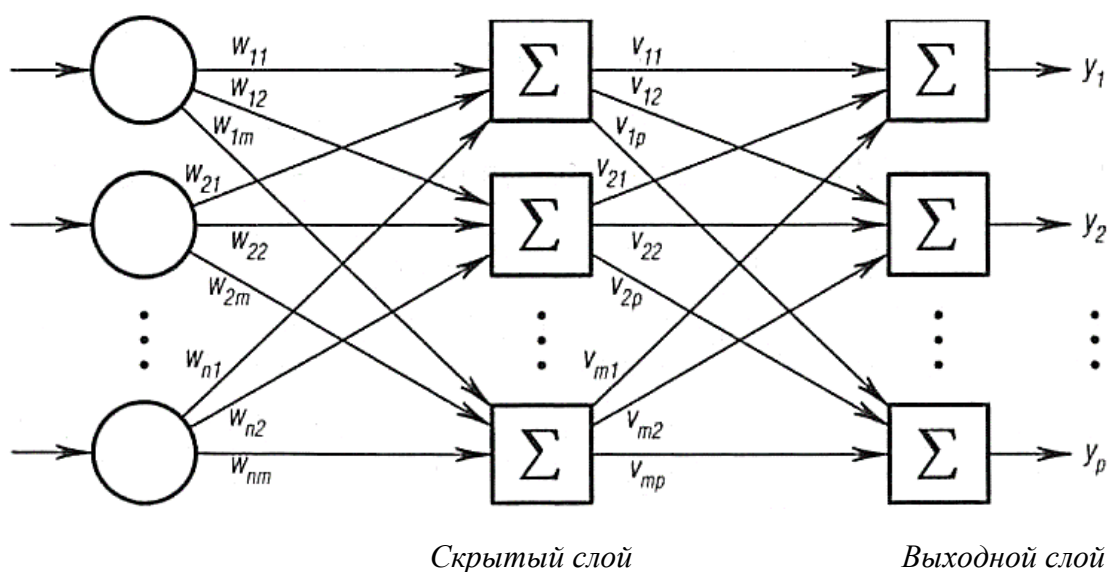


Рисунок 2. ИНС обратного распространения

Матрица весовых коэффициентов от входов к скрытому слою — W , а матрица весов, соединяющих скрытый и выходной слой — V . Обозначения для индексов: входы индекс i , элементы скрытого слоя — y , а выходы — k . Число входов сети — n , число нейронов в скрытом слое — n , число нейронов в выходном слое — p . Пусть сеть обучается на выборке $(X^t, Y^t), t=1, 2, \dots, T$.

При обучении нейронной сети ставится задача минимизации целевой функции ошибки, которая находится по методу наименьших квадратов:

$$E(W, V) = \frac{1}{2} \sum_{k=1}^p (y_k - d_k)^2,$$

где y_k — полученное реальное значение k -го выхода нейросети при подаче на нее одного из входных образов обучающей выборки; d_k — требуемое значение k -го выхода для этого образца.

Обучение нейросети производится известным оптимизационным методом градиентного спуска, т. е. на каждой итерации производится следующее изменение веса

$$w_{ij}^{N+1} = w_{ij}^N - \alpha \frac{\partial E}{\partial w_{ij}}, v_{jk}^{N+1} = v_{jk}^N - \alpha \frac{\partial E}{\partial v_{jk}}$$

где α — параметр, определяющий скорость обучения.

В качестве активационной функции в сети обратного распространения обычно используется сигмоидальная функция

$$f(s) = \frac{1}{1 + e^{-s}},$$

где s — взвешенная сумма входов нейрона. Эта функция удобна для вычислений в градиентном методе, так как имеет простую производную:

$$f'(s) = \frac{e^{-s}}{(1 + e^{-s})^2} = f(s)(1 - f(s)).$$

Функция ошибки в явном виде не содержит зависимости от весовых коэффициентов w_{ij} и v_{jk} .

После упрощений, вычислений и введения обозначений

$$\delta_k = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial s_k} = (y_k - d_k) y_k (1 - y_k),$$

Получим следующие выражения для производных:

$$\frac{\partial E}{\partial v_{jk}} = \delta_k y_j^c, \frac{\partial E}{\partial w_{ij}} = \left(\sum_{k=1}^p \delta_k v_{jk} \right) y_j^c (1 - y_j^c) x_i.$$

Алгоритм обучения сети обратного распространения

Рассмотрим полный алгоритм обучения нейросети:

Шаг 1. Инициализация сети.

Весовым коэффициентам присваиваются малые случайные значения, например, из диапазона $(-0,3; 0,3)$; задаются ε — параметр точности обучения, α — параметр скорости обучения (как правило, $\alpha \approx 0,1$ и может еще уменьшаться в процессе обучения), N_{max} — максимально допустимое число итераций.

Шаг 2. Вычисление текущего выходного сигнала.

На вход сети подается один из образов обучающей выборки и определяются значения выходов всех нейронов нейросети.

Шаг 3. Настройка синаптических весов.

Рассчитать изменение весов для выходного слоя нейронной сети по формулам

$$v_{jk}^{N+1} = v_{jk}^N - \alpha \frac{\partial E}{\partial v_{jk}}, \text{ где } \frac{\partial E}{\partial v_{jk}} = \delta_k y_j^c, \delta_k = (y_k - d_k) y_k (1 - y_k),$$

Рассчитать изменение весов для скрытого слоя по формулам

$$w_{ij}^{N+1} = w_{ij}^N - \alpha \frac{\partial E}{\partial w_{ij}}, \text{ где } \frac{\partial E}{\partial w_{ij}} = \left(\sum_{k=1}^p \delta_k v_{jk} \right) y_j^c (1 - y_j^c) x_i.$$

Шаг 4. Шаги 2-3 повторяются для всех обучающих векторов. Обучение завершается по достижении для каждого из обучающих образов значения функции ошибки, не превосходящего ε или после максимально допустимого числа итераций — N_{max} .

Замечание 1. На шаге 2 векторы из обучающей последовательности лучше предъявлять на вход в случайном порядке.

Замечание 2. Желательно наделять каждый нейрон обучаемым смещением. Это позволяет сдвигать начало отсчета логистической функции, с эффектом, аналогичном подстройке порога персептронного нейрона, и приводит к ускорению процесса обучения. Эта возможность может быть легко введена в обучающий алгоритм с помощью добавления к каждому нейрону дополнительного входа. Его вес обучается так же, как и все остальные веса, за исключением того, что подаваемый на него сигнал всегда равен +1, а не выходу нейрона предыдущего слоя.

Замечание 3. Количество входов и выходов сети, как правило, диктуется условиями задачи, а размер скрытого слоя находят экспериментально.

Обычно число нейронов в нем составляет 30—50% от числа входов. Слишком большое количество нейронов скрытого слоя приводит к тому, что сеть теряет способность к обобщению (она просто запоминает элементы обучающей выборки и не реагирует на схожие образцы, что неприемлемо для задач распознавания). Если же число нейронов в скрытом слое слишком мало, сеть оказывается не в состоянии обучиться.

Замечание 4. Выходы каждого нейрона сети лежат в диапазоне $(0, 1)$ — области значений логистической функции — это надо учитывать при формировании обучающей выборки. Если необходимо получить от сети бинарный выход, то, как правило, вместо 0 используют 0,1, а вместо 1 — 0,9, так как границы интервала недостижимы.

Модификации алгоритма обратного распространения связаны с использованием различных функций ошибки, других активационных функций, различных процедур определения направления и величины шага.

Описание программы, ее ключевые особенности и новшества

Окунишь в 1985 год и нарисуй чёрно-белое изображение размером 14 на 10 пикселей в высоту и ширину соответственно и заставь этот ноутбук сделать что-нибудь полезное — разобрать написанную цифру! Или хотя бы попытаться.

Передовой графический интерфейс обеспечен стандартными средствами языка Python на основе модуля Tkinter. Каждый Checkbutton привязан к своему входному нейрону/персептрону сети и передаёт значение 1 или 0 в зависимости от нажатия и текущего статуса.

Командный интерфейс управления через консоль позволит изменить значение веса в каждом из слоёв нейронной сети на любой позиции не закрывая и не перекомпилирую всю программу — и сразу же удостовериться во внесённых изменениях.

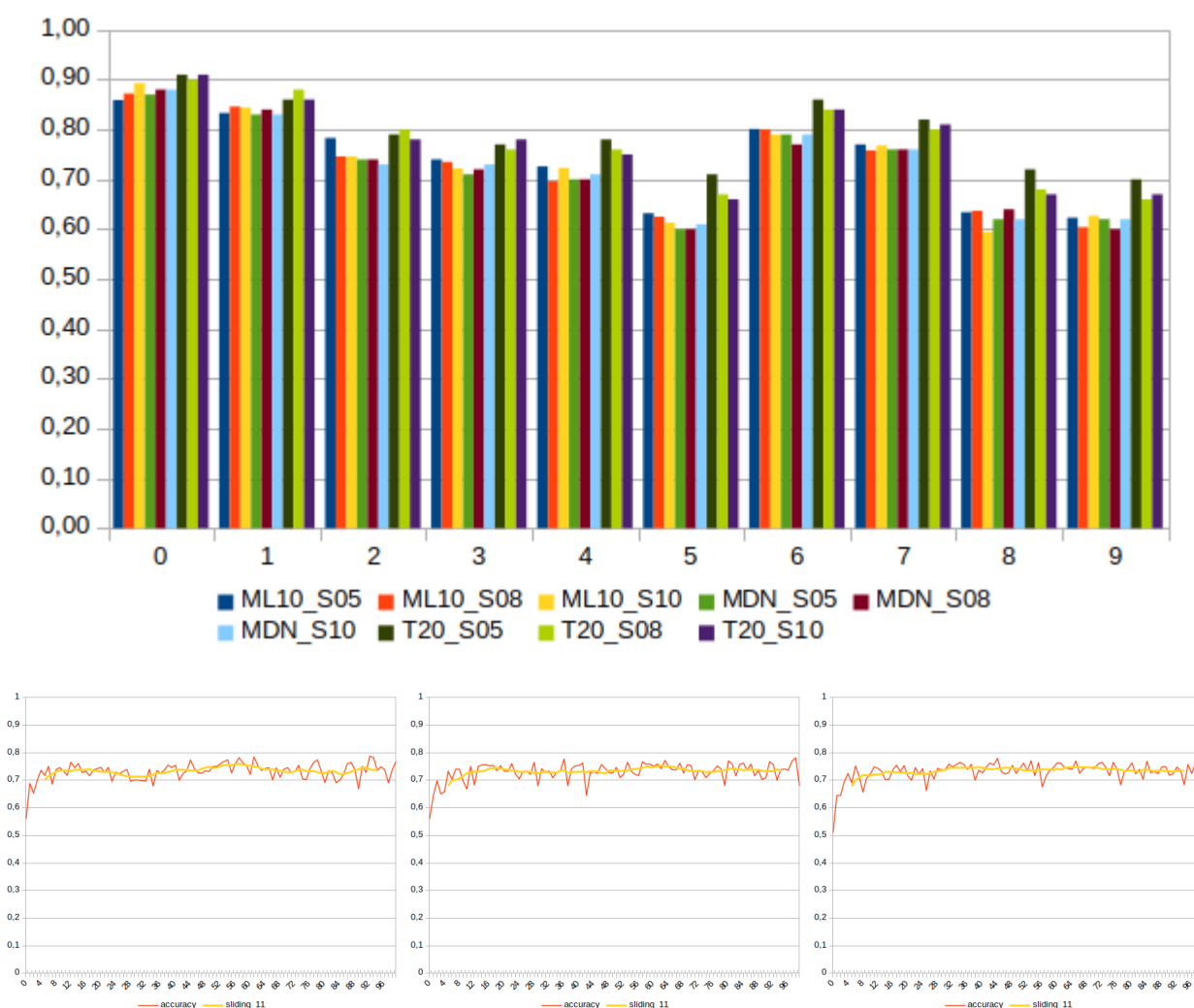
Протоколы проведенных экспериментов, с указанием погрешности распознавания

При анализе различных нейронных сетей и их активационных будем менять саму целевую функцию, либо менять скорость обучения (100 эпох).

При этом будем сравнивать между собой среднюю точность по последним 10-и эпохам (ML10), медианное значение точности по всем эпохам обучения (MDN) и верхние доверительные интервалы (T20).

Однослойная нейронная сеть.

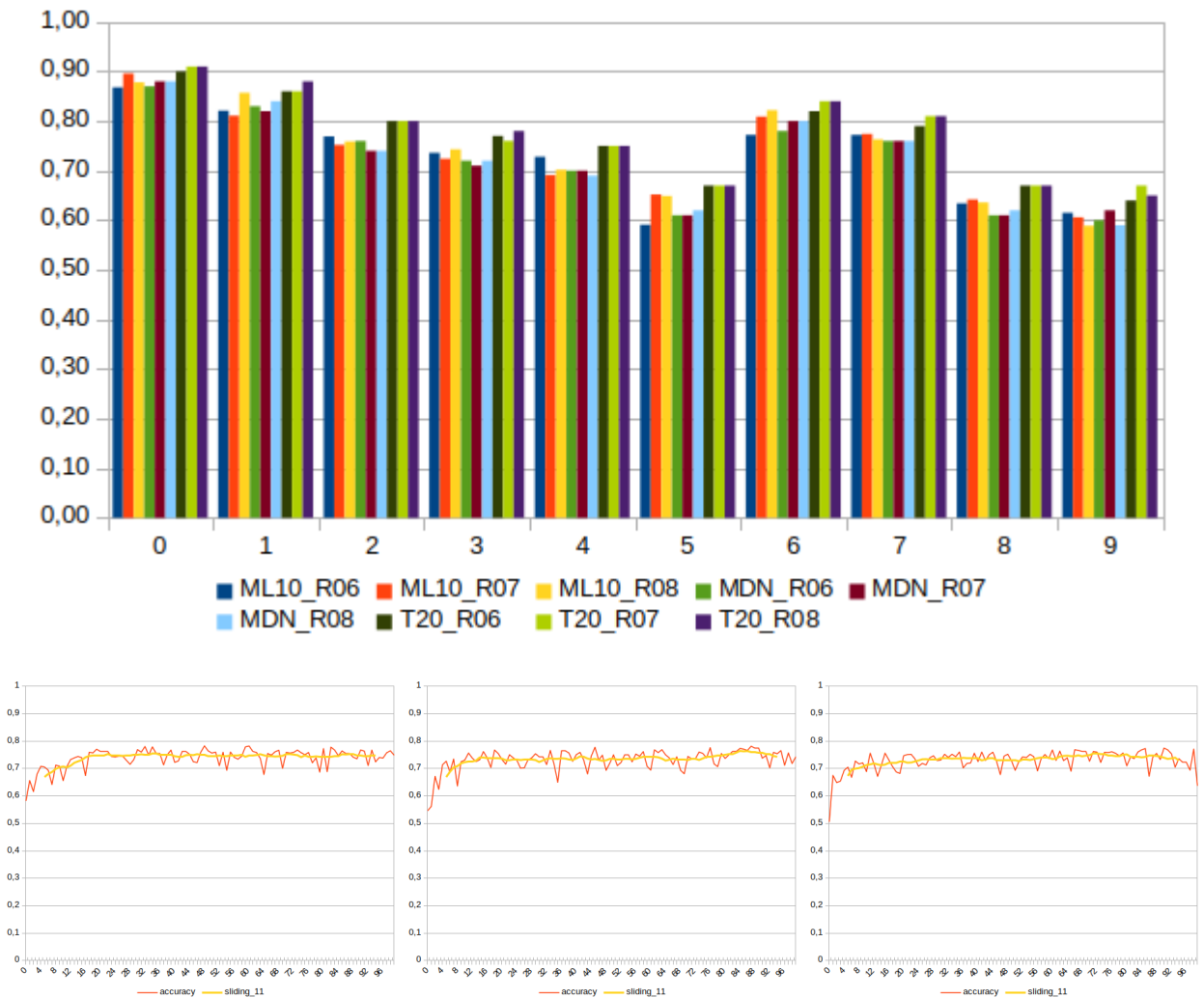
Сигмоидная активационная функция:



Для сигмоидной функции менялся её наклон, но не скорость обучения.

Точность определения быстро выходит на средний уровень в 70–75% и в остальных параметрах от наклона не зависит.

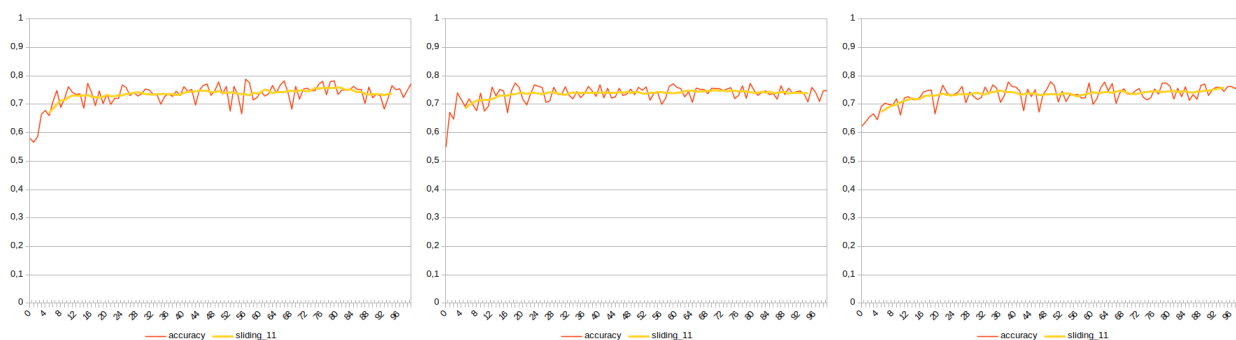
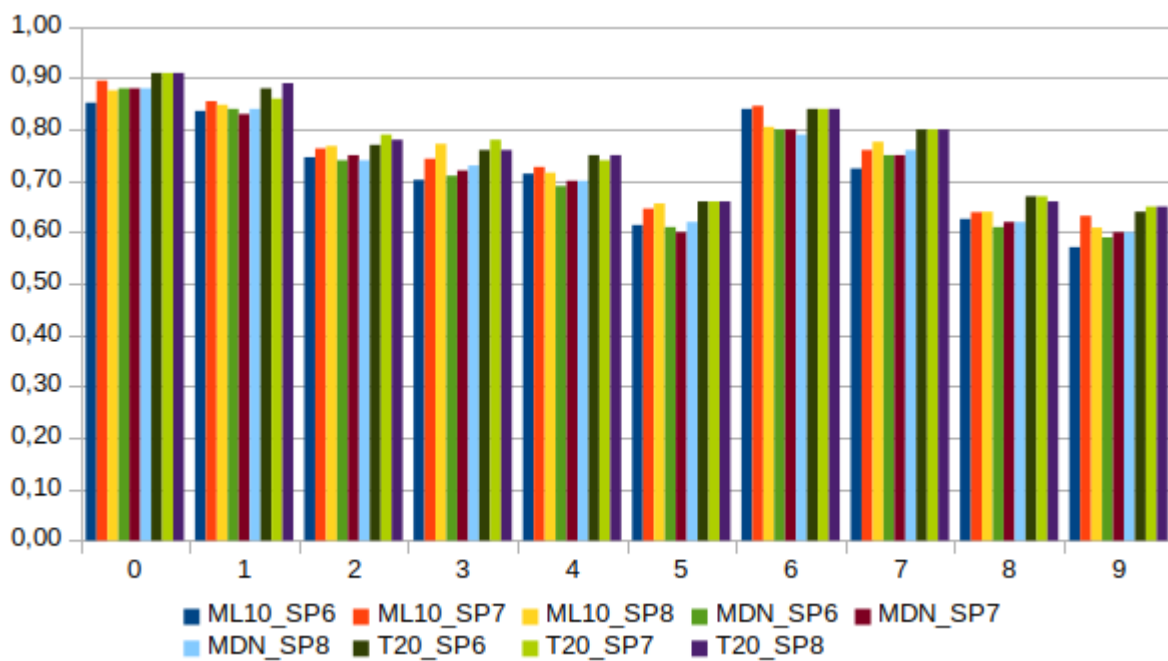
Активационная функция relu:



Для функции relu меняем скорость обучения с 0.6 до 0.8.

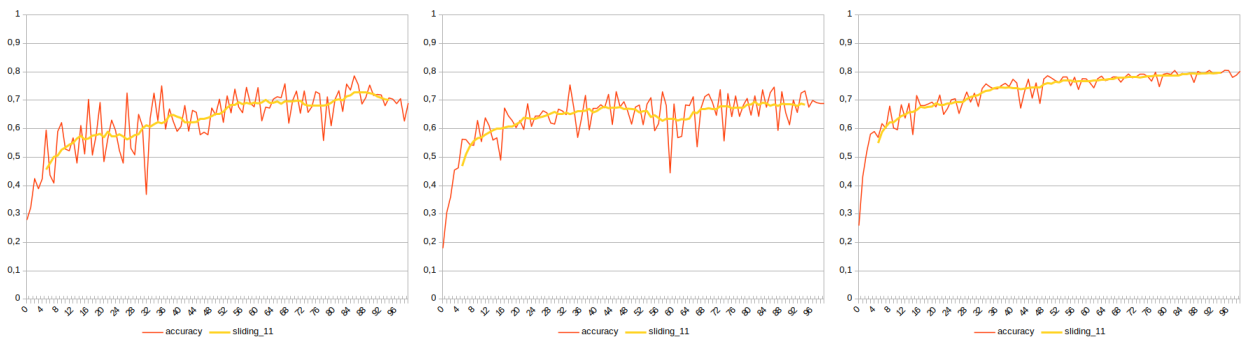
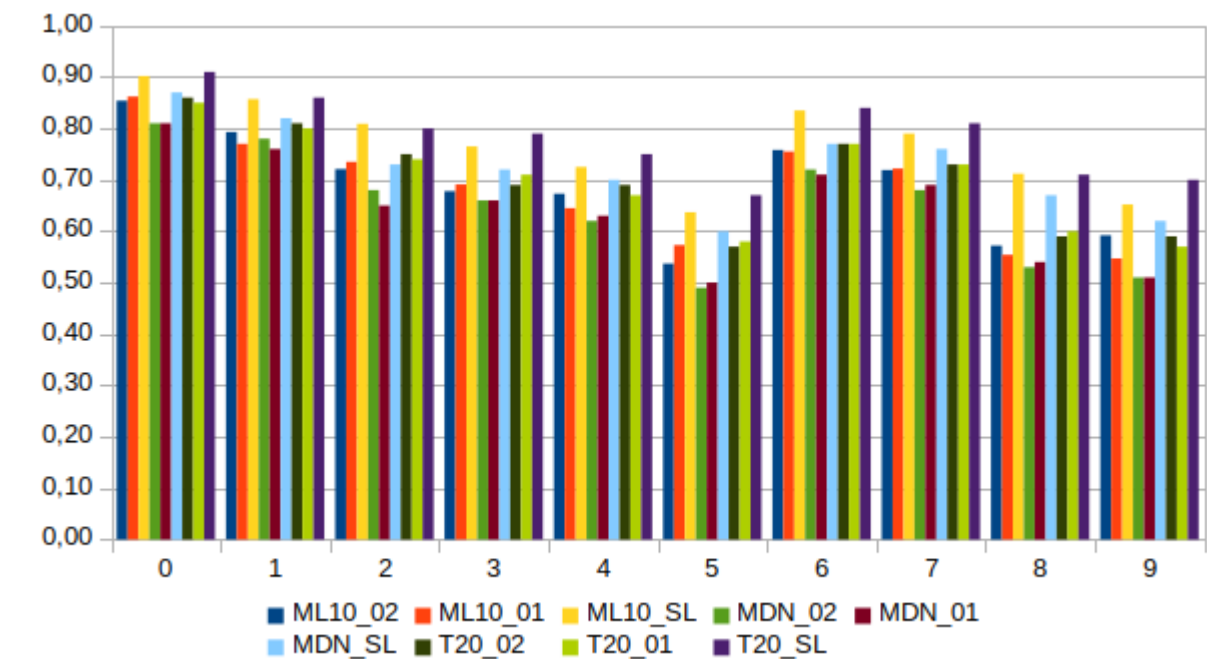
Как и с сигмоидной функцией, функция relu к 20-ой эпохе достигла стабильных значений точности (те же 70–75%), а также результаты точности для каждой цифры по отдельности повторяют результаты сигмoиды.

Активационная функция softplus:



Результаты применения функции softplus значительных изменений в результаты работы сети не внесли — получены аналогичные результаты.

Двуслойная нейронная сеть обратного распространения ошибки



Статичная скорость обучения (0.2 и 0.1 для первых двух графиков) дала посредственные результаты по сравнению с однослойной нейронной сетью (<70% против 75%), однако применение динамической скорости обучения (с постепенным уменьшением) позволило получить почти плавный переходный процесс, вырваться к 80% определительной мощности.

Выводы

В процессе лабораторной работы были рассмотрены 2 типа нейронных сетей, установлено, что однослойная ИНС позволяет задать нижний предел точности определения/классификации образов при высокой относительной производительности, когда применение нестандартных методов и большего числа слоёв может как улучшить, так и ухудшить результат при больших временных затратах на обучение.

Результаты работы однослойных нейронных сетей почти полностью повторяют друг друга независимо от активационной функции и почти сразу выходят в насыщение, когда многослойные нейронные сети долго разгоняются, но дают более стабильный результат.

Используемая литература

1. Хайкин С. Нейронные сети: полный курс. 2-е изд. Пер. С англ. М.: Издательский дом «Вильямс», 2006. 1104 с.
2. Чулюков В.А., Астахова И.Ф., Потапов А.С. и др. Системы искусственного интеллекта: Учебное пособие. Под ред. Астаховой И.Ф. – М.: БИНОМ. Лаборатория знаний, 2012. – 292 с.
3. Комарцова Л.Г., Максимов А.В. Нейрокомпьютеры. –М.:Из-во МГТУ им. Н.Э. Баумана, 2004. – 320 с.
4. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика: Пер. с англ. – М.: Мир, 1992. – 457 с.
5. Терехов В.А., Ефимов Д.В., Тюкин И.Ю. Нейросетевые системы управления: Учебное пособие для вузов. –М.: Высшая школа, 2002. – 183 с.
6. Терехов В.И. Методы вычислительного интеллекта, применяемые для решения задач инженерного и инженерно-технического обеспечения. Монография. –М.: ВУНЦ СВ «ОВА ВС РФ», 2009. – 253 с.