

Нейронные сети. Лекции.

Лабунец Л. В.

3 апреля 2024 г.

Лекция

1.1 Математическая модель многослойного перцептрона

Рассмотрим на примере 3-слойной архитектуры (в количество слоёв входят только слои, реализующие нелинейное отображение). Рассмотрим сигналы, поступающие на вход нейронов, но для избежания путаницы, будем визуализировать связи только для первых нейронов соответствующих слоёв. Рассмотрим эти связи и соотв. им параметры.



Рис. 1: Структура многослойного перцептрона

3-слойный перцептрон содержит 0-й слой - слой источников (черные квадратики). Имеется m_0 входов, на которые поступают признаки $\vec{x} = (1, x_0, \dots, x_{m_0})$, 3 скрытых слоя.

Примем следующие правила индексации скрытых нейронов и нейронов выходного слоя: первый эл-т индекса - номер нейрона, второй - номер слоя: $m_{i,j}$ - i -й нейрон j -го скрытого слоя.

Выпишем веса нейронов i -го слоя в виде матрицы W_i , столбец которой $\vec{w}_i^{(k)}$ является вектором весов i -го нейрона k -го слоя. Количество строк матрицы - количество входов слоя + 1 (на вход поступает расширенный вектор), количество столбцов - количество нейронов слоя. Элемент матрицы $w_{i,j}^{(k)}$ - вес i -го входа j -го нейрона k -го слоя. У первого скрытого слоя имеется m_0 входов (слой источников) и он содержит m_1 нейронов.

$$W_1 = \begin{pmatrix} w_{0,1}^{(1)} & w_{0,2}^{(1)} & \dots & w_{0,m_1}^{(1)} \\ w_{1,1}^{(1)} & w_{1,2}^{(1)} & \dots & w_{1,m_1}^{(1)} \\ w_{2,1}^{(1)} & w_{2,2}^{(1)} & \dots & w_{2,m_1}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m_0,1}^{(1)} & w_{m_0,2}^{(1)} & \dots & w_{m_0,m_1}^{(1)} \end{pmatrix} \quad (1)$$

Рассмотрим первый слой. На вход каждого нейрона слоя поступает сигнал смещения +1, взвешиваемый *биасом* $w_{0,i}^{(1)}$. Также на вход этого нейрона поступают входные признаки x_i , взвешиваемые *синаптическими весами* $w_{1,j}^{(1)}$ до $w_{m_0,j}^{(1)}$ - веса i -го признака j -го нейрона в первом слое. Для экономии места введём вектор синаптических весов конкретного нейрона:

$$\vec{w}_j^{(k)} = \begin{pmatrix} w_{0,j}^{(k)} & w_{1,j}^{(k)} & \dots & w_{m_0,j}^{(k)} \end{pmatrix}^T \quad (2)$$

Индекс в скобках - номер слоя, нижний индекс - номер нейрона. Т.е., $\vec{w}_1^{(1)}$ - вектор синаптических весов 1-го нейрона 1-го слоя.

На выходе i -го нейрона k -го слоя формируется сигнал $u_i^{(k)}$. Выпишем формулу для вычисления сигнала на выходе 1-го нейрона 1-го слоя $u_1^{(1)}$. Опять, в скобках - номер слоя, индекс - номер нейрона. Для этого необходимо посчитать сигнал на выходе сумматора k -го нейрона (*индуцированного локального поля*):

Не очень удачно были выбраны индексы, но оставил, как на презентации

$$v_k^{(1)} = \sum_{j=0}^{m_0} w_{j,k}^{(1)} \cdot x_j = [\vec{w}_k^{(1)}]^T \cdot \vec{x} \quad (3)$$

$$v_2^{(1)} = [\vec{w}_2^{(1)}]^T \cdot \vec{x}$$

где j - номер нейрона, k - номер слоя, $w_{j,k}^{(1)}$ - компоненты ..., \vec{x} - расширенный вектор.

Из приведённых выражений можем получить общее: $W_k^T \cdot \vec{x}$ - совокупность элементов на выходе сумматоров всех нейронов k -го слоя.

Полученные на выходе сумматора значения подвергаются нелинейному преобразованию с помощью функции активации $\varphi(W_k^T \cdot \vec{x})$. Получаем совокупность сигналов на выходе функций активации нейронов. Для первого слоя: $\varphi(W_1^T \cdot \vec{x})$, для второго: $\varphi(W_2^T \cdot \vec{x})$ и т.д.

Для получения из матрицы W_1 матрицы W_2 рисуем таблицу:

Количество входов и выходов скрытых слоёв (без учёта "расширенности" векторов):

	1 слой	2 слой	3 слой
Количество входов	m_0	m_1	m_2
Количество выходов	m_1	m_2	m_3

Выпишем матрицу W_2 :

$$W_1 = \begin{pmatrix} w_{0,1}^{(2)} & w_{0,2}^{(2)} & \dots & w_{0,m_2}^{(2)} \\ w_{1,1}^{(2)} & w_{1,2}^{(2)} & \dots & w_{1,m_2}^{(2)} \\ w_{2,1}^{(2)} & w_{2,2}^{(2)} & \dots & w_{2,m_2}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m_1,1}^{(2)} & w_{m_1,2}^{(2)} & \dots & w_{m_1,m_2}^{(2)} \end{pmatrix} \quad (4)$$

И получим выходы сумматоров нейронов второго слоя:

Починить расширенные вектора!

$$\vec{u}_2 = W_2^T \cdot \left[\frac{1}{\varphi(W_1^T \cdot \vec{x})} \right] \quad (5)$$

В формуле 1 в якобы числителе - добавленный первый элемент, чтобы получить расширенный вектор-столбец.

$$\vec{u}_3 = \varphi(W_3^T \left[\frac{1}{\varphi(W_2^T \cdot \left[\frac{1}{\varphi(W_1^T \cdot \vec{x})} \right])} \right]) \quad (6)$$

Для 4-го слоя получим:

Выписать матрицу с рисунка для 4 слоя, поправить формулу и выписать её

Увеличение слоёв нецелесообразно из-за рекурсивной структуры модели многослойного перцептрона (последовательность вложенных блоков), что приводит к уменьшению скорости вычисления, увеличению занимаемой памяти и накопления ошибок. Также по мере увеличения параметров модели, возникает *эффект переобучения*, а по мере увеличения количества слоёв, количество параметров увеличивается экспоненциально.

Возможно применение альтернативных архитектур - например, сети радиальных базисных функций. Также необходимо применение методов регуляризации, например, метод *хирургии мозга* - исключения из каждого слоя неинформативных нейронов или целых слоёв. Существуют методы обучения, подразумевающие цикличность. Более детально см. "Нейронные сети. Полный курс" Саймона Хайкина (?).

Также существует теорема, доказанная Колмогоровым, утв., что многомерная скалярная функция может воспроизводить любую закономерность сколь угодно большой сложности. Теорема об универсальной аппроксимации многослойным перцептроном, утв., что архитектура многослойного перцептрона *теоретически* позволяет с любой заданной точностью синтезировать многомерную по количеству входов векторную функцию, что позволяет строить модели нелинейных дискриминантных границ в пространстве признаков и, соотв., определять принадлежность данных кластерам.

1.2 Системный подход к синтезу алгоритмов обучения многослойного перцептрона

Рисунок *hyperrflat 5 1*

Системный подход к синтезу алгоритмов обучения многослойного перцептрона

1. Выбор цели обучения;
2. Оценка градиента средних потерь целевой функции по параметрам нейронной сети;
3. Использование алгоритмом поиска оптимальных параметров.

Предполагается наличие функционала качества работы системы, оптимальное значение которого и ищется.

Будем рассматривать 3 объекта анализа

1. Пространство входных воздействий \vec{x} - сигналы, поступающие на вход системы (обучающие и контрольные примеры), и соответствующее пространство входных воздействий, в котором существует область D все возможных входных воздействий;
2. Вектор параметров системы \vec{C} и пространство, ему соответствующее, в котором существуют эквипотенциальные гиперповерхности, все точки которых соотв. определенным значениям функционала качества. Необходимо найти область мимальных/максимальных значений этого функционала качества;
3. Функция стоимости (потерь, приобретений) $Q(\vec{x}, \vec{C})$, определяющая потери/приобретения, которые можно измерить, если на вход системы поступает воздействие \vec{x} , а система находится в состоянии \vec{C} .

Набор этих 3 объектов позволяет математически строго сформулировать выбор цели обучения в виде функционала средних потерь (среднего риска). Сделать это несложно, необходимо протестировать область D всех входных воздействий: зафиксировать \vec{x} и выбрать параметры системы \vec{C} . Для каждого зафиксированного выбора параметров системы необходимо вычислить потери по выбранной для каждого входного воздействия и просуммировать их:

$$J(\vec{C}) = \int_D Q(\vec{x}, \vec{C}) p(\vec{x}) d\vec{x} \quad (7)$$

Полученное $J(\vec{C})$ - средний риск (средние потери), $p(\vec{x})$ - весовая функция, сглаживающая аномальные значения (плотность распределения вероятностей входных воздействий).

На практике возможны 2 ситуации:

1. Нетипичный случай полной априорной определённости - полное знание входных воздействий (весовой функции $p(\vec{x})$, Типа распределения и тд.);
2. Типичный случай - функция задана набором обучающих примеров

Во втором случае необходимо выполнить предварительную выборочную оценку весовой функции $p(\vec{x})$ по набору обучающих примеров.

Рисунок hyperflat 4 1

Однако, будем считать весовую функцию $p(\vec{x})$ известной, тогда возможно посчитать функционал ошибок $J(\vec{C})$.

Алгоритм обучения нейронной сети (метод градиентного поиска):

1. Выбрать начальное приближение параметров нейронной сети $\vec{C}[0]$, 0 - номер итерации, и оценить потери;
2. Выбрать оптимальное направление поиска (вектор антиградиента $-\nabla \vec{C}$, указывающий направление, в котором потери убывают с максимальной скоростью);
3. Выбрать шаг поиска (используется процедура стохастической аппроксимации Роббенса-Монро, в соответствии с которой с каждой итерацией длина шага уменьшается обратно пропорционально номеру итерации);

В полученной точке выполняется подтверждение сходимости (сравнение итераций), вычисление нового направления и шага.

Процедура описывается следующим выражением:

$$\vec{C}[i] = \vec{C}[i-1] - \Gamma[i] \nabla \vec{C} \{J(\vec{C}[i-1])\} \quad (8)$$

Цель обучения:

$$\vec{C}_{opt} = \arg \min_{\vec{C}} \{J(\vec{C})\} \quad (9)$$

Существуют также методы, не требующие вычисления частной производной, - метод деформируемого многогранника Нелдера-Мида (симплекс-поиск).

Прикладное нелинейное программирование, Химмельблау

Обучение сводится к решению задачи оптимизации. Существуют ограничения на значения параметров системы в виде равенств или неравенств.

Результаты выполнения алгоритма зависят от выбора начального приближения, что приводит к необходимости выбора нескольких начальных приближений для обнаружения всех локальных минимумов. Существенное преимущество в этом случае обеспечивают генетические алгоритмы.

1.3 Алгоритм обучения нейронной сети для случая набора обучающих примеров