§1. Принципы Нейронных сетей.

1.1 Нейрон и принцип его работы.

В 1943 году Уорреном МакКаллоком (Warren McCulloch) и Уолтером Питтсом (Walter Pitts) была предложена модель искусственного нейрона. За основу был взят принцип работы биологического нейрона. По сути, нейрон - это реализация скалярной функции векторного аргумента. Математическая модель нейрона:

$$y = f(\sum_{i=1}^{n} w_i x_i)$$

где S-результат суммирования, w_i - веса передаваемых значений, x_i - входные данные, f - нелинейное преобразование (функция активации). Можно провести аналогию с биологическим нейроном подобным образом: входные данные - это импульсы, которые поступают в нейрон, веса - интенсивность сигнала.

В целом входной сигнал и весовые коэффициенты являются вещественными числами, а во многих конкретных задачах принадлежат конкретному множеству чисел. Выходной сигнал нейрона определяется функцией активации. В модели МакКаллока-Питтса использовалась функция Хевисайда, однако потом были предложены другие типы функций активации, которые обеспечивали более плавное изменение выходного сигнала нейрона.

МакКаллок и Питтс предложили также метод объединения отдельных нейронов в искусственные нейронные сети. Для этого выходные сигналы нейрона передаются на вход следующему нейрону . Нейронная сеть состоит из нескольких слоев, на каждом из которых может находиться несколько нейронов. Слой, который принимает сигналы из внешнего мира, называется входным. Слой, который выдает сигналы во внешний мир, — выходным.

Остальные слои называются скрытыми. Входной слой нейронных элементов выполняет распределительные функции, выходной слой служит для обработки информации от предыдущих слоев и выдачи результата. И выходной, и скрытые слои являются обрабатывающими. Выход каждого нейронного элемента предыдущего слоя нейронной сети соединен со входами нейронных элементов следующего слоя.

1.2 Различные виды функций активации

Для разных задач используют различные функции активации. К примеру, для прогнозирования часто используют линейную функцию активации $y = w_i x_i + b$

2. Обучение нейронных сетей

Обучение нейронной сети — это процесс определения весов соединений между нейронами таким образом, чтобы сеть приближала необходимую функцию с заданной точностью. Существует три подхода к обучению нейронных сетей [3]: обучение с учителем (supervised learning), обучение без учителя (unsupervised learning) и обучение с подкреплением (reinforcement learning). При обучении с учителем на вход сети подаются наборы входных сигналов (объектов), для которых заранее известен правильный ответ (обучающее множество). Веса меняются по определенным правилам в зависимости от того, правильный ли выходной сигнал выдала сеть. При обучении без учителя на вход сети подаются объекты, для которых правильный выходной сигнал заранее не известен. Обучение с подкреплением предполагает наличие внешней среды, с которой взаимодействует сеть. Обучение происходит на основании сигналов, полученных от этой среды.

Нейронные сети МакКаллока—Питтса не обучались. Веса для всех входов нейронов должны были быть заданы заранее.

Впервые идею обучения нейронных сетей предложил Дональд Хэбб (Donald Hebb) в 1949 году [19]. Согласно Хэббу, связи нейронов, которые активируются вместе, должны усиливаться, а связи нейронов, которые срабатывают отдельно друг от друга, должны ослабевать. Хэбб предложил правила изменения веса входных сигналов нейронов в соответствии с тем, правильный ответ выдавала сеть, или нет [19] (обучение с учителем). А.В. Новиков доказал сходимость предложенного метода обучения нейрона на основе правил Хэбба [20], при условии, что выборка объектов линейно разделима. Впоследствии было предложено несколько аналогичных правил как для обучением с учителем [21–23], так и без учителя [24–28].

В 1970 году А.Г. Ивахненко разработал метод группового учета аргументов (group method of data handling) [29, 30], позволяющий не только вычислять веса связей между нейронами, но и определять количество слоев в сети и нейронов в них в зависимости от потребностей прикладной задачи. Используя подход обучения с учителем, уровни сети инкрементально строятся и обучаются на основе обучающего множества с использованием регрессионного анализа. Затем происходит этап упрощения сети с применением отдельного множества объектов с известными правильными ответами, которое не использовалось при обучении (проверочное множество, validation set). Для исключения ненужных нейронов из сети используется регуляризация. В работе [30] описано применение метода группового учета аргументов для обучения глубокой нейронной сети, состоящей из восьми слоев. Метод широко использовался на практике [31–33].

В настоящее время для обучения нейронных сетей, в том числе глубоких, используется алгоритм обратного распространения ошибки (error backpropagation algorithm), основанный на методе градиентного спуска. Алгоритм был предложен в 1970 году в магистерской диссертации [34, 35] без связи с нейронными сетями. Первое применение этого алгоритма для обучения нейронных сетей описано в работе [36], вышедшей в 1981 году. После этого появилось еще несколько работ на эту тему [37–39]. Алгоритм обратного распространения ошибки использует обучение с учителем, для него требуется обучающее множество с заранее известными правильными ответами. Вводится мера ошибки, которая определяет, насколько сильно выходные значения сети отличаются от правильных ответов. Затем мера ошибки минимизируется с помощью метода градиентного спуска путемизменения значений весов в сети. Для того чтобы оценить, насколько сильно каждый вес влияет на выходное значение, рассчитываются частные производные ошибки по весам. Затем производится изменение весов на небольшие значение с учетом градиента. Так повторяется до тех пор, пока ошибка на выходе не сократится до допустимых значений. Начальные значения весов нейронов в сети задаются случайным образом.

В глубокой нейронной сети с несколькими скрытыми слоями производится расчет ошибки, которая передается от одного слоя к другому. На первом этапе рассчитывается значение ошибки на выходе нейронной сети, для которого мы знаем правильные ответы. Затем рассчитывается ошибка на входе в выходной слой сети, которая будет использоваться как ошибка на выходе скрытого слоя. Таким способом расчет продолжается до того момента, когда будет известна ошибка на входном слое. Именно поэтому

алгоритм имеет название обратное распространение ошибки.

Возможно несколько вариантов реализации обучения нейронных сетей с помощью алгоритма обратного распространения ошибки. При полном обучении градиент рассчитывается для всех объектов обучающего множества. Однако такой подход часто не является эффективным в случае, когда обучающее множество большое и для обработки всех его элементов требуется значительное время. Альтернативный вариант — использование метода стохастического градиентного спуска, при котором веса изменяются при обработке одного элемента обучающего множества (онлайн-обучение) или нескольких элементов (обучение на пакетах или мини-выборках). На практике для обучения нейронных сетей чаще всего используется именно метод стохастического градиентного спуска или его модификации [40–42].

3. Архитектура глубоких нейронных сетей

Нейронная сеть, показанная на рис. 2, называется полносвязной. В такой сети каждый нейрон следующего слоя связан со всеми нейронами предыдущего слоя. Однако это не единственный вариант соединения нейронов в сеть. В данном разделе рассматривается развитие архитектур нейронных сетей.

В 1980 году Кунихико Фукушима (Kunihiko Fukushima) предложил архитектуру нейронной сети, которая называется неокогнитрон [43]. Архитектура использовала аналогию со сложными и простыми клетками в зрительной коре кошки [44]. Простые клетки срабатывают в ответ на простые визуальные сигналы, такие как ориентация границ. Сложные клетки менее зависимы от пространственного расположения сигналов и ориентируются на более общие признаки. В неокогнитроне простым клеткам

соответствуют сверточные слои (convolutional layers), а сложным — слои подвыборки (subsampling layers). В сверточных слоях окно сверточного узла (convolutional unit) с заданным набором весов (ядро свертки) перемещается по двумерному массиву входных данных, например, по пикселям изображения (рис. 3). Значения совпадающих элементов в данных и ядре свертки перемножаются, полученные результаты складываются и поступают в нейрон следующего слоя. Все сверточные узлы используют одинаковые ядра свертки, поэтому для описания сверточной сети требуется относительно немного параметров. Как правило, в сверточных слоях используется не одно, а несколько ядер свертки.