Приложение № 2
к распоряжению РТУ МИРЭА
от__ ____________№__________

Титульный лист материалов по дисциплине
(заполняется по каждому виду учебного материала)

ДИСЦИЛИНА Проектирование и обучение нейронных сетей
(полное наименование дисциплины без сокращений)

ИНСТИТУТ Информационые технологии

КАФЕДРА Вычислительная техника
полное наименование кафедры

ВИД УЧЕБНОГО МАТЕРИАЛА Лекция
(в соответствии с пп.1-11)

ПРЕПОДАВАТЕЛЬ Сорокин Алексей Борисович
(фамилия, имя, отчество)

СЕМЕСТР 7 семестр, 2023/2024
(указать семестр обучения, учебный год)

1. ЛЕКЦИЯ. Понятие о математическом нейроне

Биологический прототип

Развитие искусственных нейронных сетей вдохновляется биологией. То есть, рассматривая сетевые конфигурации и алгоритмы, исследователи применяют термины, заимствованные из принципов организации мозговой деятельности. Но на этом аналогия заканчивается. Наши знания о работе мозга столь ограничены, что мало бы нашлось точно доказанных закономерностей для тех, кто пожелал бы руководствоваться ими. Поэтому разработчикам сетей приходится выходить за пределы современных биологических знаний в поисках структур, способных выполнять полезные функции. Во многих случаях это приводит к необходимости отказа от биологического правдоподобия, мозг становится просто метафорой, и создаются сети, невозможные в живой материи или требующие неправдоподобно больших допущений об анатомии и функционировании мозга.

Несмотря на то, что связь с биологией слаба и зачастую несущественна, искусственные нейронные сети продолжают сравнивать с мозгом. Их функционирование часто имеет внешнее сходство с человеческим познанием, поэтому трудно избежать этой аналогии. К сожалению, такие сравнения неплодотворны и создают неоправданные ожидания, неизбежно ведущие к разочарованию.

Нервная система человека, построенная из элементов, называемых нейронами, имеет ошеломляющую сложность. Около 10^{11} нейронов участвуют в примерно 10^{15} передающих связях, имеющих длину метр и более. Каждый нейрон обладает многими свойствами, общими с другими органами тела, но ему присущи абсолютно уникальные способности: принимать, обрабатывать и передавать электрохимические сигналы по нервным путям, которые образуют коммуникационную систему мозга.

Нейро́н (от др.-греч. – волокно, нерв) — структурно-функциональная единица нервной системы, представляющая собой электрически возбудимую клетку, которая обрабатывает, хранит и передает информацию посредством электрических и химических сигналов (рис.1).

Аксон – это нейрит (длинный цилиндрический отросток нервной клетки), по которому нервные импульсы идут от тела клетки (сомы) к другим нервным клеткам.

Дендрит – разветвлённый отросток нейрона, который получает информацию через химические (или электрические) синапсы от аксонов других нейронов и передаёт её через электрический сигнал телу нейрона, из которого вырастает.

Си́ напс (греч. — соединение, связь) — место контакта между двумя нейронами.

Рис.1. Биологический нейрон

Дендриты идут от тела нервной клетки к другим нейронам, где они принимают сигналы в точках соединения, называемых синапсами. Принятые синапсом входные сигналы передаются к телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, другие — воспрепятствовать его возбуждению.

Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону сигнал другим нейронам. У этой основной функциональной схемы много усложнений и исключений, тем не менее, большинство искусственных нейронных сетей моделируют лишь эти простые свойства.

Проведя аналогию с биологическим можно построить искусственный нейрон (рис. 2).

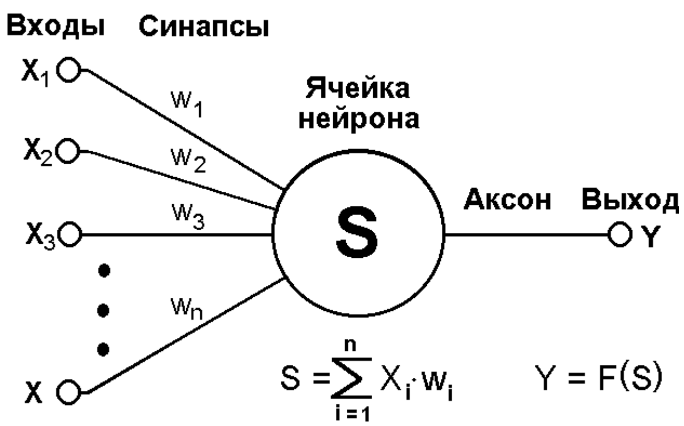


Рис.2. Искусственный (математический) нейрон

Наиболее часто нейронные сети используются для решения следующих задач:

Классификация образов – указание принадлежности входного образа, представленного вектором признаков, одному или нескольким предварительно определенным классам.

Кластеризация – классификация образов при отсутствии обучающей выборки с метками классов.

Прогнозирование – предсказание значения $y(t_{n+1})$ при известной последовательности $y(t_1), y(t_2), \dots, y(t_n)$.

Оптимизация – нахождение решения, удовлетворяющего системе ограничений и максимизирующим или минимизирующим целевую функцию.

Память, адресуемая по содержанию (ассоциативная память) –память, доступная при указании заданного содержания.

Управление – расчет такого входного воздействия на систему, при котором она следует по желаемой траектории.

Математический нейрон Мак-Каллока – Питтса.

Первой работой, которая заложила теоретический фундамент для создания интеллектуальных устройств, моделирующих человеческий мозг на самом низшем—структурном—уровне, принято считать опубликованную в 1943 г. статью Уоррена Мак-Каллока и Вальтера Питтса «*Идеи логических вычислений в нервной деятельности*». Они предложили математическую модель нейрона мозга человека, назвав ее ***математическим*** или ***модельным нейроном***.

Авторы математического нейрона предложили изображать нейрон в виде кружочка со стрелочками (рис.3).

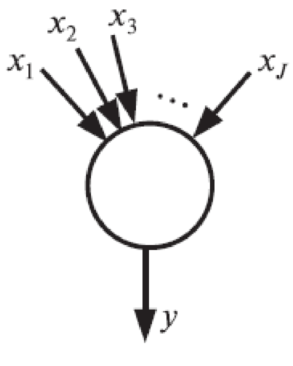


Рис.3. Одно из изображений искусственного нейрона

Стрелки означают входы и выход нейрона. Через входы математический нейрон принимает входные сигналы $\mathbf{x_1, x_2, \dots, x_j, \dots, x_J}$ и суммирует их, умножая каждый входной сигнал на некоторый весовой коэффициент $\mathbf{w_j}$:

$$S = \sum_{j=1}^{J} w_j x_j$$

После выполнения операции суммирования математический нейрон формирует выходной сигнал $\boldsymbol{y}$ согласно следующему правилу:

$$y = \begin{cases} 1, & если \quad S \geq \theta \\ 0, & если \quad S < \theta \end{cases}$$

где θ – порог чувствительности нейрона.

Таким образом, математический нейрон может существовать в двух состояниях. Если взвешенная сумма входных сигналов ***S*** меньше порога ***θ***, то его выходной сигнал ***y*** равен нулю. В этом случае говорят, что нейрон не возбуждён. Если же входные сигналы достаточно интенсивны и их взвешенная сумма достигает порога чувствительности ***θ***, то нейрон переходит в возбуждённое состояние, и на его выходе, согласно правилу, образуется сигнал ***y=1***.

Иногда встречается следующее схематичное изображение (рис.4), которое более точно отображает процессы происходящие в математическом нейроне.

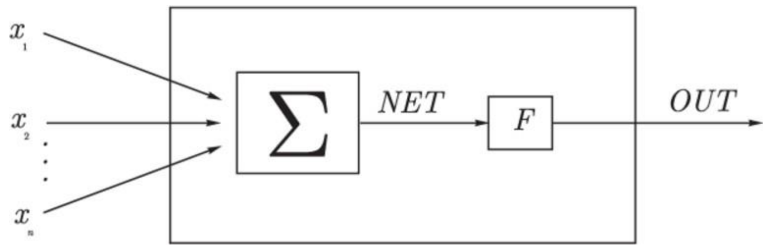


Рис.4. Схематичное изображение

Тогда входные сигналы ($\mathbf{x_1, x_2, \ldots, x_n}$), можно в совокупности обозначить вектором ***X***, а веса $(\mathbf{w_1, w_2, \ldots, w_n})$ совокупностью **W**. Выход из суммирующего блока обозначается ***NET***.

$$\boldsymbol{NET = S = X}\mathbf{W}$$

Сигнал ***NET*** далее преобразуется активационной функцией ***F*** и дает выходной нейронный сигнал ***OUT***. Активационная функция может быть обычной линейной функцией

$$\boldsymbol{OUT = F(NET)}$$

где ***F*** – константа, пороговой функцией

$$OUT = \begin{cases} 1, & \text{если } NET > T; \\ 0, & \text{если } NET \leqslant T \end{cases}$$

где ***T*** – некоторая постоянная пороговая величина, или же функция, более точно моделирующая нелинейную передаточную характеристику биологического нейрона и предоставляющей нейронной сети большие возможности.

1. Активационная (пороговая) функция

Активационная функция нейрона определяет нелинейное преобразование, осуществляемое нейроном и вычисляющая выходной сигнал формального нейрона. Выбор активационной функции определяется спецификой поставленной задачи либо ограничениями, накладываемыми некоторыми алгоритмами обучения.

Существует множество видов активационных функций, но более всего распространены следующие три:

1. ПОРОГОВАЯ ФУНКЦИЯ.

Другое название — ***функция Хевисайда***. Представляет собой перепад. До тех пор пока взвешенный сигнал на входе нейрона не достигает некоторого уровня $\boldsymbol{T}$ – сигнал на выходе равен нулю. Как только сигнал $\boldsymbol{NET}$ на входе $\boldsymbol{F}$ превышает указанный уровень выходной сигнал $\boldsymbol{OUT}$ скачкообразно изменяется на единицу (рис.5).

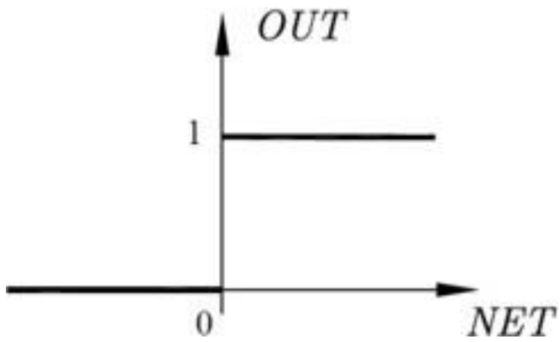


Рис.5. Пороговая функция

Математическую запись функции я вам уже приводил:

$$OUT = \begin{cases} 1, \text{если } NET \geq T \\ 0, \text{если } NET < \mathrm{T} \end{cases}$$

Таким образом, математический нейрон представляет собой пороговый элемент с несколькими входами и одним выходом. Каждый математический нейрон имеет свое определенное значение порога чувствительности $\boldsymbol{T}$.

Пример расчета с порогом.

Авторы математического нейрона в своей статье также показали, что с помощью математического нейрона с помощью пороговой функции можно моделировать различные логические функции, например функцию логического умножения ***«И» («AND» конъюнкция),*** функцию логического сложения ***«ИЛИ» («OR»)*** и функцию логического отрицания ***«НЕТ» («NOT»)***.

Таблицы истинности логических функций.

И (∧)		
x_1	x_2	$x_1 \wedge x_2$
0	**0**	**0**
0	**1**	**0**
1	**0**	**0**
1	**1**	**1**

ИЛИ (∨)		
x_1	x_2	$x_1 \vee x_2$
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**1**

НЕТ(¬)	
x	$\neg x$
0	**1**
1	**0**

Представим математические нейроны (рис.6)

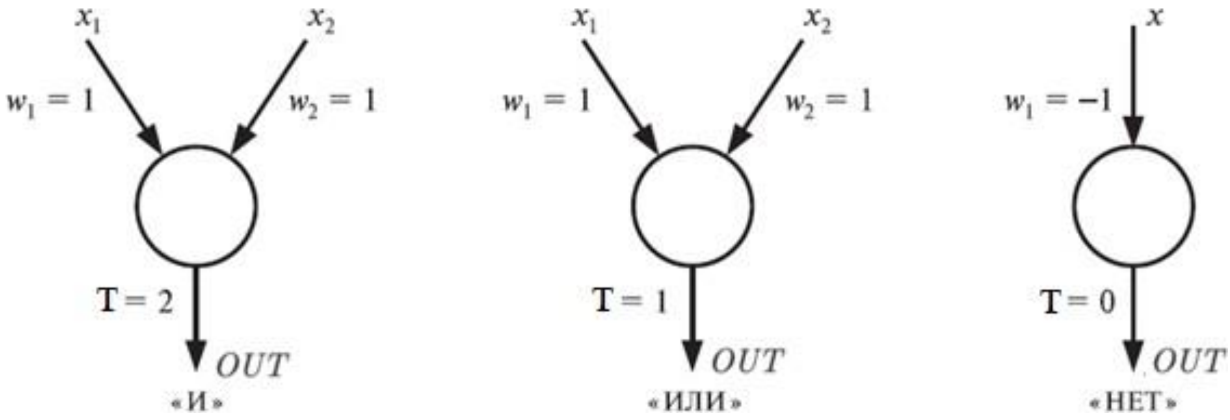


Рис.6. Математические нейроны для «И», «ИЛИ» и «НЕТ»

Рассчитываем ***логическое «И»***. Сигнал ***NET*** может быть :

При $\boldsymbol{x_1 = 0,\ x_2 = 0 \rightarrow x_1 * w_1 + x_2 * w_2 = 0*1 + 0*1 = NET_1 = 0}$

При $\boldsymbol{x_1 = 0,\ x_2 = 1 \rightarrow x_1 * w_1 + x_2 * w_2 = 0*1 + 1*1 = NET_2 = 1}$

При $\boldsymbol{x_1 = 1,\ x_2 = 0 \rightarrow x_1 * w_1 + x_2 * w_2 = 1*1 + 0*1 = NET_3 = 1}$

При $\boldsymbol{x_1 = 1,\ x_2 = 1 \rightarrow x_1 * w_1 + x_2 * w_2 = 1*1 + 1*1 = NET_4 = 2}$

Устанавливаем пороговую функцию

$$OUT = \begin{cases} 1, \text{если } NET \geq 2 \\ 0, \text{если } NET < 2 \end{cases}$$

Тогда на выходе: $\boldsymbol{OUT_1 = 0,\ OUT_2 = 0,\ OUT_3 = 0,\ OUT_4 = 1}$

Рассчитываем ***логическое «ИЛИ»*. Сигналы *NET*** может такие же как у *логического «И»,* но пороговая иная.

$$OUT = \begin{cases} 1, \text{если } NET \geq 1 \\ 0, \text{если } NET < 1 \end{cases}$$

Тогда на выходе $\boldsymbol{OUT_1 = 0,\ OUT_2 = 1,\ OUT_3 = 1,\ OUT_4 = 1}$

Рассчитываем ***логическое «НЕТ»*.** Тогда сигнал ***NET*** может быть :

При $\boldsymbol{x = 0 \rightarrow x * w = 0*(-1) = NET_1 = 0}$

При $\boldsymbol{x = 1 \rightarrow x * w = 1*(-1) = NET_2 = -1}$

Устанавливаем пороговую функцию для логического отрицания.

$$OUT = \begin{cases} 1, \text{если } NET \geq 0 \\ 0, \text{если } NET < 0 \end{cases}$$

Тогда на выходе: $\boldsymbol{OUT_1 = 1,\ OUT_2 = 0}$

Таким образом, если рассматривать пороговую функцию, то в ней происходит смещение на величину ***T*** (рис.7).

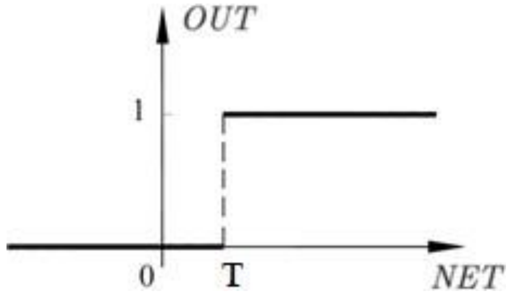


Рис.7. Пороговая функция со смещением

Пороговая функция может иметь и другую интерпретацию (рис.8).

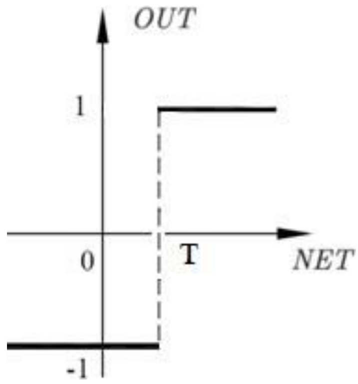


Рис.8. Пороговая функция симметричный вид

Математическую запись функции будет выглядеть:

$$OUT = \begin{cases} 1, \text{если } NET \geq T \\ -1, \text{если } NET < T \end{cases}$$

Поэтому для упрощения расчета в современном исполнении нейрона вводят понятия смещения ***b,*** которое отличается от ***T*** только знаком: ***b* = - *T***. При этом нейронное смещение ***b*** можно рассматривать как вес $\boldsymbol{w_0}$ некоторого дополнительного входного сигнала $\boldsymbol{x_0 = 1}$, величина которого всегда равна единице.

$$\mathbf{NET = S} = \sum_{j=1}^{J} \boldsymbol{w_j x_j} + \boldsymbol{w_0\, x_0}$$

Таким образом, дополнительный вход $\boldsymbol{x_0}$ и соответствующий ему вес $\boldsymbol{w_0}$ используются для ***инициализации нейрона***. Под инициализацией подразумевается смещение активационной функции нейрона по горизонтальной оси, то есть формирование порога чувствительности нейрона.

Тогда пороговая активационная функция нейрона примет вид (рис.9):

$$OUT = \begin{cases} 1, \text{если } NET \geq 0 \\ 0, \text{если } NET < 0 \end{cases}$$

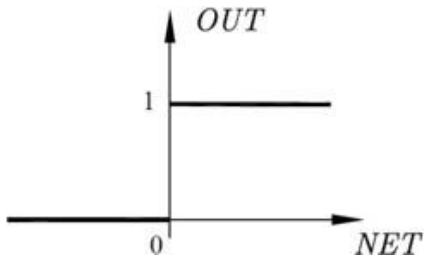


Рис.9. Пороговая функция для смещения

Или более симметричный вид (рис.10)

$$OUT = \begin{cases} 1, \text{если } NET \geq 0 \\ -1, \text{если } NET < 0 \end{cases}$$

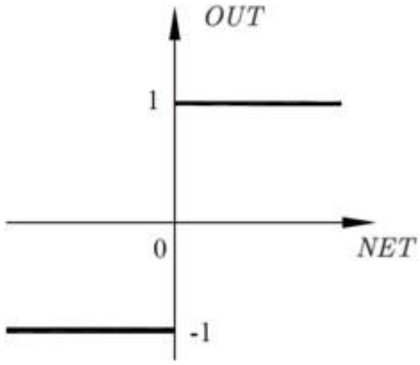


Рис. 10. Симметричный вид для смещения

Пример расчета со смещением.

<u>Рассчитываем ***логическое «И»***</u> (рис.11)

Установим $\boldsymbol{x_0 = 1}$, а $\boldsymbol{w_0 = -3/2}$

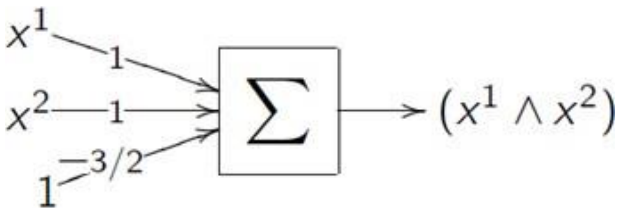


Рис. 11. Математический нейрон для логического «И»

При $x_1=0,\ x_2=0 \rightarrow x_1*w_1 + x_2*w_2 + x_o*w_0 = 0*1 + 0*1 - 1*3/2 = NET_1 = -1,5$

При $x_1=0,\ x_2=1 \rightarrow x_1*w_1 + x_2*w_2 + x_o*w_0 = 0*1 + 1*1 - 1*3/2 = NET_2 = -0,5$

При $x_1=1,\ x_2=0 \rightarrow x_1*w_1 + x_2*w_2 + x_o*w_0 = 1*1 + 0*1 - 1*3/2 = NET_3 = -0,5$

При $x_1=1,\ x_2=1 \rightarrow x_1*w_1 + x_2*w_2 + x_o*w_0 = 1*1 + 1*1 - 1*3/2 = NET_4 = 0,5$

Знаем, что пороговая функция для смещения равна

$$OUT = \begin{cases} 1, \text{если } NET > 0 \\ 0, \text{если } NET \leq 0 \end{cases}$$

Тогда на выходе: $\boldsymbol{OUT_1 = 0,\ OUT_2 = 0,\ OUT_3 = 0,\ OUT_4 = 1}$

Пороговая функция делит входное векторное пространство на две части гиперплоскостью $\boldsymbol{(x_1 + x_2 = 1,5)}$, классифицируя входные векторы как относящиеся к первому классу, если выходной сигнал ***OUT* > *0*** или ко второму классу, если выходной сигнал ***OUT* ≤ *0*** (рис. 12).

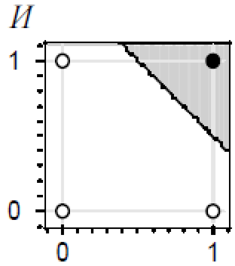


Рис. 12. Гиперплоскость для логического «И»

Рассчитываем ***логическое «ИЛИ»*** (рис.13)

Установим $\boldsymbol{x_0 = 1}$, а $\boldsymbol{w_0 = -1/2}$

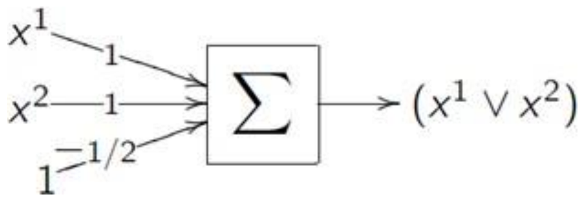


Рис. 13. Математический нейрон для логического «ИЛИ»

При $x_1=0,\ x_2=0 \rightarrow x_1*w_1 + x_2*w_2 + x_o*w_0 = 0*1 + 0*1 - 1*1/2 = NET_1 = -0,5$

При $x_1=0,\ x_2=1 \rightarrow x_1*w_1 + x_2*w_2 + x_o*w_0 = 0*1 + 1*1 - 1*1/2 = NET_2 = 0,5$

При $x_1=1,\ x_2=0 \rightarrow x_1*w_1 + x_2*w_2 + x_o*w_0 = 1*1 + 0*1 - 1*1/2 = NET_3 = 0,5$

При $x_1=1,\ x_2=1 \rightarrow x_1*w_1 + x_2*w_2 + x_o*w_0 = 1*1 + 1*1 - 1*1/2 = NET_4 = 1,5$

Тогда согласно активационной функции на выходе получаем $\boldsymbol{OUT_1 = 0,}$ $\boldsymbol{OUT_2 = 1,\ OUT_3 = 1,\ OUT_4 = 1}$

Графическая интерпретация будет следующей при гиперплоскости $\boldsymbol{(x_1 + x_2 = 0,5)}$ (рис.14)

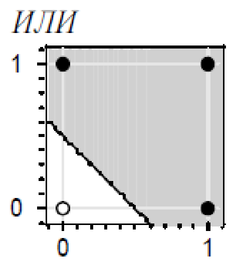


Рис. 14. Гиперплоскость для логического «И»

Аналогично рассчитываем логическое отрицание при $\boldsymbol{x_0 = 1}$, а $\boldsymbol{w_0 = 1/2}$ (рис.15)

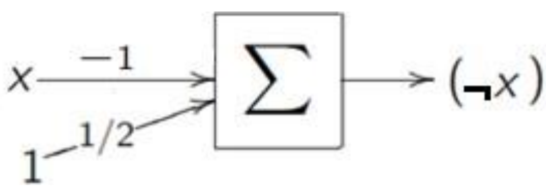


Рис. 15. Математический нейрон для логического «НЕТ»

При $x = 0 \rightarrow x^* w + x_0^* w_0 = 0^*(-1) + 1^* 1/2 = NET_1 = 1/2$

При $x = 1 \rightarrow x^* w + x_0^* w_0 = 1^*(-1) + 1^* 1/2 = NET_2 = -1/2$

Тогда на выходе: $\boldsymbol{OUT_1 = 1, OUT_2 = 0}$

Графическая интерпретация будет следующей при гиперплоскости ***(x = 0,5)*** (рис.16)

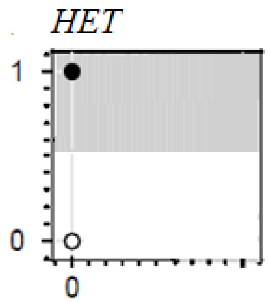


Рис. 16. Гиперплоскость для логического «НЕТ»

Сигмоидальная функция или сигмоид

Если блок ***F*** сужает диапазон изменения величины ***NET*** так, что при любых значениях ***NET*** значения ***OUT*** принадлежат некоторому конечному интервалу, то ***F*** называется «сжимающей» функцией. В качестве «сжимающей» функции часто используется ***"сигмоидальная"*** (S-образная) функция, показанная. Часто под сигмоидой понимают логистическую функцию (рис.17).

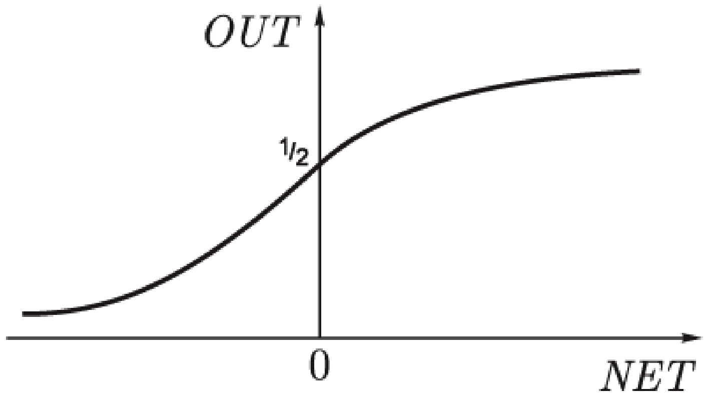


Рис. 17. Сигмоид

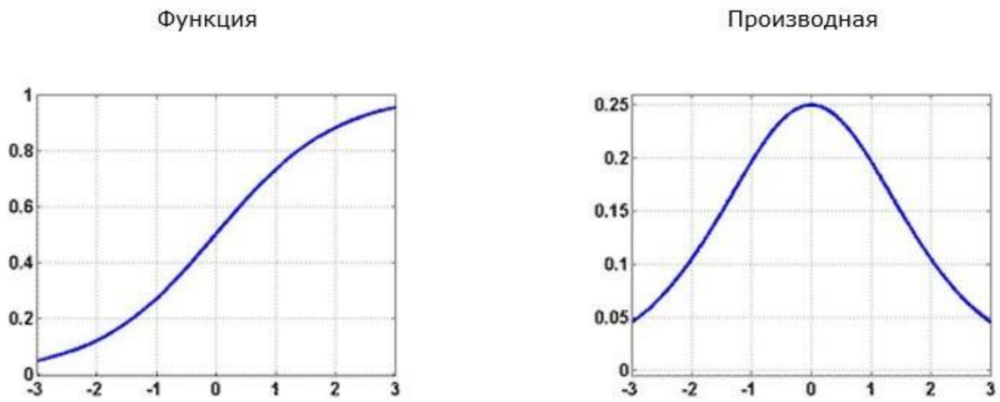
Эта функция математически выражается как

$$F(x) = {}^{1}/_{(1 + e^{-x})}$$

Таким образом

$$OUT = {}^{1}/_{(1 + e^{-NET})}$$

Одна из причин, по которой сигмоид используется в нейронных сетях, это простое выражение его производной через саму функцию (которое и позволило существенно сократить вычислительную сложность *метода обратного распространения ошибки, сделав его применимым на практике*).(рис. 18)



$$f(n) = \frac{1}{1 + \exp(-n)} \qquad f(n) = \frac{exp(-n)}{(1 + exp(-n))^2}$$

Рис. 18. Функция и производная

где $\exp = e^n$ (экспонента)

Тогда

$$\frac{\partial OUT}{\partial NET} = OUT(1 - OUT)$$

Его дополнительное преимущество состоит в автоматическом контроле усиления. Для слабых сигналов (т.е. когда ***OUT*** близко к нулю) кривая вход-выход имеет сильный наклон, дающий большое усиление. Когда величина сигнала становится больше, усиление падает. Таким образом, большие сигналы воспринимаются сетью без насыщения, а слабые сигналы проходят по сети без чрезмерного ослабления. Другими словами центральная область логистической функции, имеющая большой коэффициент усиления, решает проблему обработки слабых сигналов, в то время как области с падающим усилением на положительном и отрицательном концах подходят для больших возбуждений.

Гиперболический тангенс

Другой широко используемой активационной функцией является *гиперболический тангенс* (рис. 19). По форме она сходна с логистической функцией и часто используется биологами в качестве математической модели активации нервной клетки. В качестве активационной функции искусственной нейронной сети она записывается следующим образом:

$$OUT = th(x).$$

Подобно логистической функции гиперболический тангенс является ***S-образной функцией***, но он симметричен относительно начала координат, и в точке ***NET = 0*** значение выходного сигнала ***OUT*** равно нулю. В отличие от логистической функции, гиперболический тангенс принимает значения различных знаков, и это его свойство применяется для целого ряда сетей.

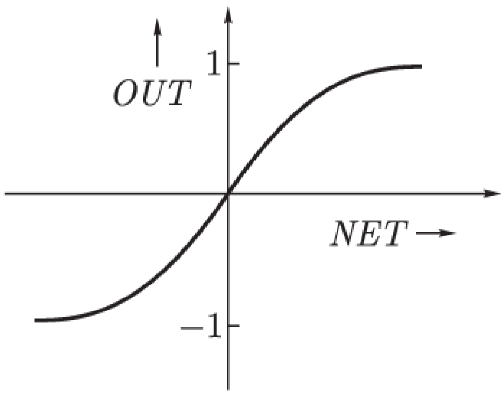


Рис. 19. Функция и производная

Линейная функция активации

Линейная функция представляет собой прямую линию и пропорциональна входу (то есть взвешенной сумме на этом нейроне) (рис. 20).

$$A = cx$$

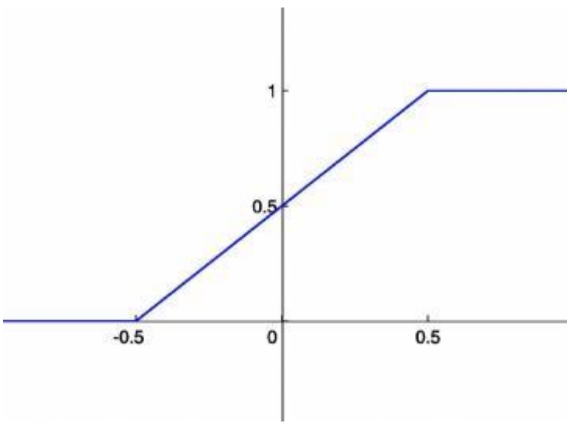


Рис. 20. Линейная функция

Такой выбор активационной функции позволяет получать спектр значений, а не только бинарный ответ. Можно соединить несколько нейронов вместе и, если более одного нейрона активировано, решение принимается на основе применения операции max (или softmax).

Однако (вы знакомы с методом градиентного спуска), и можете заметить, что для этой функции производная равна постоянной.

Производная от $A = cx$ по x равна c. Это означает, что градиент никак не связан с X. Градиент является постоянным вектором, а спуск производится по постоянному градиенту. Если производится ошибочное предсказание, то изменения, сделанные обратным распространением ошибки, тоже постоянны и не зависят от изменения на входе.

При этом каждый слой активируется линейной функцией. Значение с этой функции идет в следующий слой в качестве входа, второй слой считает взвешенную сумму на своих входах и, в свою очередь, включает нейроны в зависимости от другой линейной активационной функции.

Не имеет значения, сколько слоев мы имеем. Если все они по своей природе линейные, то финальная функция активации в последнем слое будет просто линейной функцией от входов на первом слое. Это означает, что два слоя (или N слоев) могут быть заменены одним слоем. Мы потеряли возможность делать наборы из слоев.

Полулинейный элемент

Активационная функция ReLu (англ. Rectified linear unit - Полулинейный элемент или линейный выпрямитель) имеет вид (рис.21)

$$A(x) = max(0, x)$$

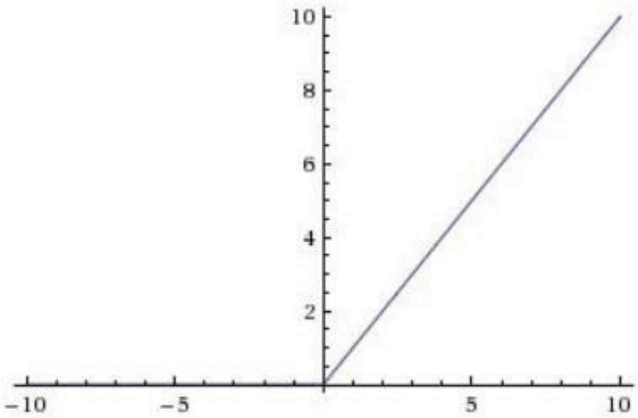


Рис. 21. Полулинейный элемент

ReLu возвращает значение x, если x положительно, и 0 в противном случае.На первый взгляд кажется, что ReLu имеет все те же проблемы, что и линейная функция, так как ReLu линейна в первом квадранте. Но на самом деле, ReLu нелинейна по своей природе. (На самом деле, такая функция является хорошим аппроксиматором, так как любая функция может быть аппроксимирована комбинацией ReLu). Область допустимых значений ReLu — [0,inf), то есть активация может “взорваться”.

Разреженность активации. Представим большую нейронную сеть с множеством нейронов. Использование сигмоиды или гиперболического тангенса будет влечь за собой активацию всех нейронов аналоговым способом. Это означает, что почти все активации должны быть обработаны для описания выхода сети. Другими словами, активация плотная, а это затратно. В идеале мы хотим, чтобы некоторые нейроны не были активированы, это сделало бы активации разреженными и эффективными.

ReLu позволяет это сделать. Представим сеть со случайно инициализированными весами (или нормализированными), в которой примерно 50% активаций равны 0 из-за характеристик ReLu (возвращает 0 для отрицательных значений x). В такой сети включается меньшее количество нейронов (разреженная активация), а сама сеть становится легче. Отлично, кажется, что ReLu подходит нам по всем параметрам. Но ничто не безупречно, в том числе и ReLu.

Из-за того, что часть ReLu представляет из себя горизонтальную линию (для отрицательных значений X), градиент на этой части равен 0. Из-за равенства нулю градиента, веса не будут корректироваться во время спуска. Это означает, что пребывающие в таком состоянии нейроны не будут реагировать на изменения в ошибке/входных данных (просто потому, что градиент равен нулю, ничего не будет меняться). Такое явление называется проблемой умирающего ReLu. Из-за этой проблемы некоторые нейроны просто выключатся и не будут отвечать, делая значительную часть нейросети пассивной. Однако существуют вариации ReLu, которые помогают эту проблему избежать.

ReLu менее требовательно к вычислительным ресурсам, чем гиперболический тангенс или сигмоида, так как производит более простые математические операции. Поэтому имеет смысл использовать ReLu при создании глубоких нейронных сетей.