Конспект лекции

А.Н. Шевляков

Введение в нейронные сети

Задача классификации для нейронных сетей

# Постановка задачи

Есть множество объектов *М* с известными значениями признака *Y*.

Найти (предсказать, оценить) значение признака *Y* для нового объекта *А*. Признак Y называется **целевым**.

Предсказываемый признак Y может быть:

* количественным;
* меткой класса.

В первом случае задача предсказания называется **задачей регрессии**,

а во втором случае – **задачей классификации**. Каждая из этих задач требует особой архитектуры нейронной сети.

**Примеры**:

* Предсказать пол человека по его фото (физическим данным).
* Предсказать наличие (отсутствие) болезни по результатам анализа.
* Предсказать результат командной многопользовательской игры по характеристикам и истории аккаунтов участников.
* Распознать тип дорожного знака для беспилотного автомобиля.
* По заданному тексту выяснить, сгенерирован он искусственным интеллектом или написан человеком.

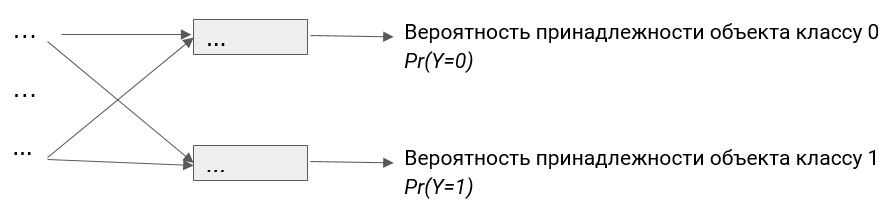
В нашем курсе мы предполагаем, что все объекты для задачи предсказания описываются набором числовых признаков.

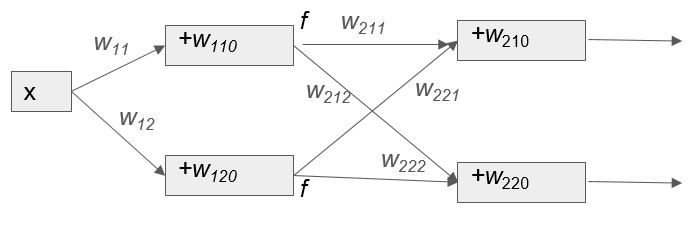
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Объект | Признак Р1 | ... | Признак Рn | Целевой признак Y |
| А1 | *p11* | ... | *p1n* | *y1* |
| А2 | *p21* | ... | *p2n* | *y2* |
| ... | ... | ... | ... | ... |
| Аm | *pm1* | ... | *pmn* | *ym* |

Иными словами, тренировочную выборку можно представить в виде таблицы. Для задачи классификации столбец Y содержит **метки класса**. Будем их обозначать с помощью натуральных чисел {0,1, 2...}.

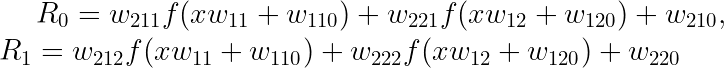
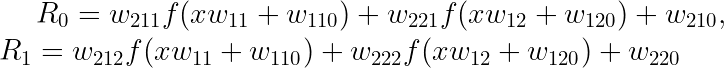
**Важно помнить:** не всё, что обозначается числами, является числом.

**Метки класса – это не числа!** Поэтому решать задачу классификации как задачу регрессии плохо!

Как выглядит выходной слой НС, решающей задачу классификации? Для простоты будем далее считать, что **классификация у нас** **бинарная**, то есть существует всего два (0,1) возможных значения целевого признака Y. Тогда НС должна заканчиваться слоем из двух нейронов.



Для объекта *х* выходы сети равны



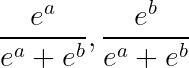
Проблема в том, что числа, выдаваемые НС, могут не вполне годиться на роль вероятностей.

Они могут быть:

* не попадать в интервал [0,1];
* их сумма может не быть равной 1.

Значит, нужно провести «умное» преобразование над числами, выдаваемыми НС.

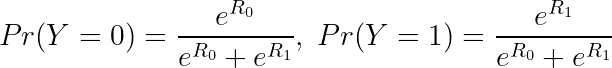
Пусть *a, b*- значения, которые выдал последний слой НС (не забывайте: у нас бинарная классификация).

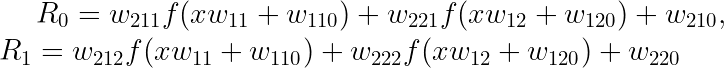
Тогда числа

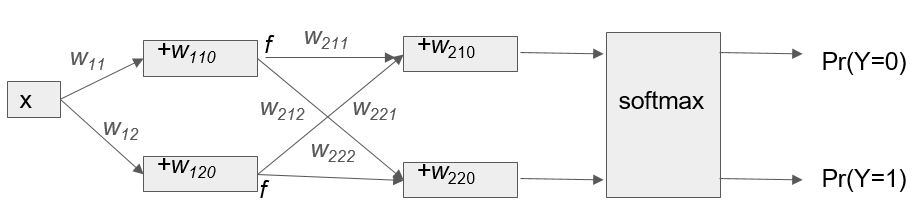
положительны и их сумма равна 1. То есть их можно интерпретировать как вероятности принадлежности классам.

Операцию перехода от чисел *a, b*к указанным выше выражениям называется операцией **softmax**.

Нужно к выходам сети *R1, R2* применить softmax. То есть сеть выдает вероятности



****

Фактически преобразование softmax является дополнительным (и последним) слоем НС, решающей задачу классификации.

**План решения задачи бинарной классификации** (полная аналогия с задачей регрессии):

1. Взять тренировочную выборку, то есть набор объектов с известными значениями целевого признака *Y*. Нейронная сеть должна восстановить зависимость между нецелевыми признаками и целевым признаком.
2. Задать основные параметры нейронной сети: количество слоёв, количество нейронов на каждом слое, тип связи между слоями и т.д.
3. Выписать выражения для вероятностей *Pr(Y=0), Pr(Y=1)*. Эти выражения будут содержать вхождения весов *wi*.
4. По ТВ составим функцию потерь *L(w)* (допустим, что мы уже умеем это делать).
5. Функция потерь *L(w)* содержит вхождения букв *wi* (весов НС). Относительно этих переменных мы находим точку минимума функции *L(w).*
6. Точка минимума определяет оптимальные веса НС.
7. Присваиваем весам НС найденные оптимальные значения.Пусть теперь объект *А* не принадлежит тренировочной выборке. Мы прогоняем *А* через НС и на выходе получаем вероятности *Pr(Y=0), Pr(Y=1) –* они и являются предсказанием для объекта *А*.

# Функция потерь для классификации

Очевидно, что нужно минимизировать выражение Pr(Y=0) для всех объектов ТВ из класса 1, и одновременно минимизировать выражение Pr(Y=1) для всех объектов ТВ из класса 0.

Или: максимизировать вероятность принадлежности к истинному классу для каждого объекта из ТВ.

Пусть ТВ состоит из объектов *x1,...,xm*, для которых известны их точные метки классов *y1 ,...,ym ,* и НС даёт вероятности *p1 ,...,pm* принадлежности **к истинному классу**.Тогда в качестве функции потерь в задаче классификации берут выражение



(это **кросс-энтропия** распределений *y1,...,ym* и *p1,...,pm*)

Насколько логично это выражение?

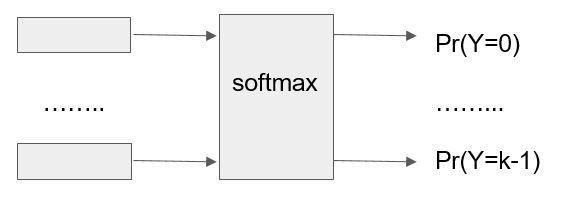
L(w)=\ln p_1+\ln p_2 +\ldots+ \ln p_mВо-первых, минимизация функции потерь *L(w)* означает максимизацию выражения

Это означает, что нужно максимизировать аргументы логарифмов *p1 ,..., pm*. Следовательно, нужно максимизировать вероятности принадлежности к истинному классу.

# Многоклассовая классификация

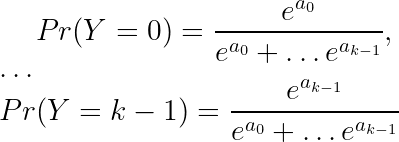
До этого классификация была бинарной. А если классов больше, чем 2?

Всё аналогично! Архитектура сети имеет ***k* выходов** (*k* – число классов).



В конце применяется **операция softmax**.

Ожидаемая формула:



где *a0 ,...,ak-1*– выходы из нейронов последнего слоя.

Для каждого объекта *Ai* ТВ можно вычислить величину *pi –* вероятность принадлежности к **истинному классу**.

Как и раньше, функция потерь равна



А дальше всё, как обычно: минимизируем эту функцию с помощью ГС.

Всё очень просто: мы же хотим, чтобы «вероятность *pi* принадлежности истинному классу» у объекта *Ai* из ТВ была максимальной?

Значит, **нужно максимизировать все числа *pi.***То есть нужно максимизировать произведение .

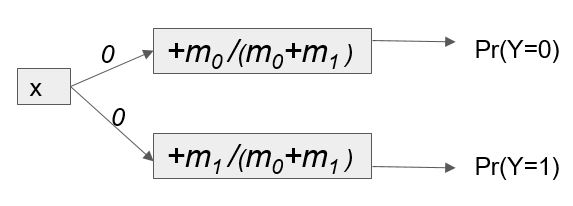
Максимизировать длинное произведение трудно, поэтому переходят к **максимизации логарифмов:**

Ну а максимизировать сумму логарифмов – это всё равно что умножить выражение на -1 и минимизировать. И мы получаем известную функцию потерь.

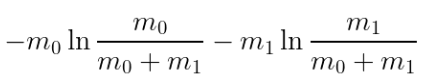
# Задача классификации

**Дурацкий классификатор.** Он работает так. Берёт ТВ: в ней m объектов, *m0* из них принадлежат классу *0* и *m1* принадлежат классу *1* (*m0+m1=m*). И абсолютно для любого объекта классификатор выдает вероятности:

Pr(Y=0)=\frac{m_0}{m_0+m_1}, \; Pr(Y=1)=\frac{m_1}{m_0+m_1},

Такой классификатор может быть реализован с помощью НС (даже без слоя softmax):

Можно вычислить значение функции потерь дурацкого классификатора



Эта величина позволяет грубо оценить силу вашей НС.

Пусть ТВ состоит из 10 объектов: 6 объектов принадлежат классу 0, 4 объекта принадлежат классу 1. Дата-саентист Вася натренировал НС такую, что значение функции потерь на ТВ равно 6.75.

**Задание**: огорчите Васю.

**Решение**: НС Васи можно сравнить с дурацким классификатором, который для данной ТВ имеет следующее значение функции потерь:

