Проект

Искусственные нейронные сети в задаче классификации

С. Лагутин githubmark.png

2018 год

Содержание

1. Обзор

1.1. Классификация

Классификация объектов — одна из стандартных задач машинного обучения. Её можно описать так: имеется множество объектов, которые каким-то образом разделены на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется *обучающей выборкой*. Классовая принадлежность остальных объектов неизвестна. Требуется построить алгоритм, способный классифицировать произвольный объект (то есть указать к какому классу он относится) из исходного множества.

В машинном обучении задача классификации относится к разделу *обучения с учителем*. Существует также *обучение без учителя*, когда разделение объектов обучающей выборки на классы не задаётся, и требуется классифицировать объекты только на основе их сходства друг с другом. В этом случае принято говорить о *задачах кластеризации*.

Одним из самых простых типов классификации является *бинарная классификация*, когда различных классов всего два. Данный тип служит основой для решения более сложных задач.

1.2. Методы решения

Для решения задач классификации могут использоваться следующие методы:

- Байесовский классификатор;
- Решающие деревья;
- Логистическая регрессия;
- Искусственные нейронные сети.

Байесовский классификатор — тип алгоритмов классификации, основанный на теореме, утверждающей, что если плотности распределения каждого из классов известны, то искомый алгоритм можно выписать в явном аналитическом виде. Более того, этот алгоритм оптимален, то есть обладает минимальной вероятностью ошибок. На практике плотности распределения классов, как правило, не известны. Их приходится оценивать по обучающей выборке. В результате байесовский алгоритм перестаёт быть оптимальным, так как восстановить плотность по выборке можно только с некоторой погрешностью. В задаче бинарной классификации звуков восстановление плотности классов является плохо решаемой проблемой.

Решающие деревья — средство поддержки принятия решений, структура которого представляет собой *листья* и *ветки*. На ветках дерева записаны атрибуты, от которых зависит целевая функция, в листьях записаны значения целевой функции, а в остальных узлах — атрибуты, по которым различаются случаи. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе.

Одним из основных вопросов в реализации решающих деревьев для задачи классификации является выбор атрибутов, по которым будет осуществляться разделение данных на классы.

Логистическая регрессия — метод построения линейной разделяющей поверхности. В случае двух классов разделяющей поверхностью является гиперплоскость. В задаче бинарной классификации звуков нельзя гарантировать возможность разделения пространства параметров одной гиперплоскостью.

Искусственная нейронная сеть — это математическая модель, построенная в некотором смысле по образу и подобию сетей нервных клеток живого организма. Для решения задачи классификации может использоваться такой тип ИНС, как *многослойный перцептрон Розенблатта*. Он представляет собой передающую сеть, состоящую из генераторов сигнала трёх типов: сенсорных элементов, ассоциативных элементов и реагирующих элементов. Производящие функции этих элементов зависят от сигналов, возникающих либо где-то внутри передающей сети, либо, для внешних элементов, от сигналов, поступающих из внешней среды.

2. Решение задачи

Для решения поставленной задачи я использовал многослойный перцептрон Розенблатта.

2.1. Перцептрон

Перцептрон состоит из нескольких слоёв нейронов:

1. Входной слой, содержащий псевдо-нейроны, которые передают дальше значения *предикторов* — параметров объекта;
2. Один или несколько скрытых слоёв;
3. Выходной слой, содержащий один нейрон.

Передача сигналов (активация) нейронной сети происходит от входного слоя, через скрытые слои, к выходному слою.

net.png

Все нейроны (кроме входного слоя) имеют одинаковое строение, состоят из двух частей — сумматорной и активационной функций. Сумматорная функция определяет то, как нейрон будет использовать входящую информацию из предыдущего слоя. Активационная функция определяет реакцию нейрона, которая будет передана по всем выходным связям в следующий слой.

В качестве сумматорной функции выбрана взвешенная сумма всех входящих сигналов:

$$S = b + \sum_{j=1}^{m} x_j w_j$$

где m — количество входящих сигналов нейрона, x_j — значение, получаемое по j-ому входу, w_j — вес j-ого входа, b — некоторое смещение, изменяемое в процессе обучения. Смещение можно учитывать в сумме, если добавить в каждый слой, кроме выходного на первое место нейрон, у которого значение активации будет всегда равно 1.

Активационная функция — логистическая (сигмоидальная):

$$\sigma\left(S\right) = \frac{1}{1 + e^{-S}}$$

`sigmoid.png`

Логистическая функция является гладкой, что необходимо для работы алгоритма обучения. Кроме того, её значение можно интерпретировать как вероятность принадлежности объекта к одному из двух классов.

Обучение нейронной сети — это настройка весов для входящих связей всех нейронов, с целью получения достоверных предсказаний. Для обучения используется *алгоритм обратного распространенния ошибки*, который основывается на градиентном спуске по простанству весов в сторону уменьшения значений целевой функции ошибки.

Для оценки правдоподобности предсказаний используется *квадратичная функ-*

ция ошибки:

$$E = \frac{1}{2N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2$$

где N — количество примеров, $\hat{y}_i$ — предсказанное значение для i-ого примера, y_i — правильный ответ для него.

Для того, чтобы понять, как изменится значение функции ошибки при изменении какого-либо веса входящих сигналов нейрона, нужно взять её частную производную по этому весу.

Сначала считается изменение весов в выходном слое:

$$\Delta w_j = -\alpha \frac{\partial E}{\partial w_j}$$

где $\alpha \in \mathbb{R}$ — скорость обучения.

В векторном виде:

$$\Delta W = -\alpha \nabla_W E$$

где $\nabla_W E = \left(\frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_m}\right)$ — градиент E в точке W.

Посчитаем частную производную от функции E по j-му весу:

$$\frac{\partial E}{\partial w_j} = \frac{\partial\left(\frac{1}{2N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2\right)}{\partial w_j}$$

Так как производная суммы равна сумме производных, возьмём для простоты один пример, а после просуммируем все значения:

$$\frac{1}{2}\cdot\frac{\partial(\hat{y}-y)^2}{\partial w_j} = \frac{1}{2}\cdot\frac{\partial\left(\hat{y}-y\right)^2}{\partial \hat{y}}\cdot\frac{\partial \hat{y}}{\partial w_j} = \left(\hat{y}-y\right)\frac{\partial\sigma\left(S\right)}{\partial w_j} =$$

$$= \left(\sigma\left(S\right) - y\right)\sigma\prime\left(S\right)\frac{\partial\sum_{j=1}^{m} x_j w_j}{\partial w_j} = \left(\sigma\left(S\right) - y\right)\sigma\left(S\right)\left(1 - \sigma\left(S\right)\right)x_j$$

Итак, общая формула для j-ого веса по N примерам:

$$\frac{\partial E}{\partial w_j} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)\hat{y}_i\left(1 - \hat{y}_i\right)x_j$$

Далее полученная ошибка распространяется по ИНС в обратном порядке, от выходного слоя ко входному, изменяя веса скрытых слоёв.

Введём следующие обозначения:

- w_{jk}^l — значение j-ого веса k-ого нейрона в l-ом слое (вес связи из j-ого нейрона $l-1$ слоя в k-ый нейрон l-ого слоя);
- m_l — количество нейронов в l-ом слое;
- s_k^l — значение сумматорной функции k-ого нейрона в l-ом слое;
- a_k^l — значение активационной функции k-ого нейрона в l-ом слое;
- $\delta_k^l = \frac{\partial E}{\partial s_k^l}$ — ошибка k-ого нейрона в l-ом слое.

Зная значение ошибки δ_k^l для каждого нейрона, можно получить соответствующее изменение его весов:

$$\Delta w_{jk}^l = -\alpha \delta_k^l a_j^{l-1}$$

Посчитаем значение ошибки для нейронов выходного (L-ого) слоя. Для простоты возьмём один пример:

$$\delta_k^L = \frac{\partial E}{\partial s_k^L} = \frac{1}{2} \cdot \frac{\partial \left(\sigma\left(s_k^L\right) - y_k\right)^2}{\partial s_k^L} = \left(a_k^L - y_k\right) a_k^L \left(1 - a_k^L\right)$$

где y_k — правильный ответ для k-ого нейрона выходного слоя.

Теперь выразим ошибку нейрона на l-ом слое через ошибки на $l+1$ слое:

$$\delta_j^l = \sigma\prime\left(s_j^l\right) \sum_{k=1}^{m_{l+1}} w_{jk}^{l+1} \delta_k^{l+1} = a_j^l \left(1 - a_j^l\right) \sum_{k=1}^{m_{l+1}} w_{jk}^{l+1} \delta_k^{l+1}$$

2.2. Предикторы из записей

Для получения численных предикторов из записей звука, массив интенсивности сигналов по времени из wave-файла переводится с помощью быстрого дискретного преобразования Фурье в спектр — массив интенсивности частот.

Если посмотреть на спектры сломанных и целых мячей, можно отчётливо заметить разницу: у целых мячей имеется узкий диапазон частот, которые имеют высокую интенсивность, по сравнению с остальными частотами; у сломанных же мячей спектры имеют несколько не очень высоких пиков.

Таким образом в качестве предикторов можно использовать значения нескольких самых интенсивных частот, или, другими словами — значения частот, в которых есть пики интенсивности в спектре.

Пример спектра записи звука целого мяча:

whole.png

Пример спектра записи звука повреждённого мяча:

broken.png