Конспект лекции

А.Н. Шевляков

Введение в нейронные сети

Контроль качества натренированных нейронных сетей

# Валидационная выборка

Вы натренировали НС. Как проверить ее точность и адекватность?

НС тренируется на ТВ и старается угадать ответы для объектов из ТВ.

Где гарантия, что для новых объектов (не из ТВ) НС будет вести себя адекватно?

**Решение.**

Вначале у вас есть большая **размеченная выборка (РВ)**, то есть множество объектов с известными ответами. РВ разбиваем на две части: **тренировочную** (ТВ) и **валидационную** (ВВ) выборки в пропорции 80% и 20% (как правило).

Как и ранее, с помощью ТВ мы тренируем НС, а по ВВ мы проверяем точность и адекватность построенной НС.

**Формулы для точности НС существенно зависит от типа задачи:**

* задача регрессии,
* задача классификации.

В любом случае все объекты ВВ нужно «пропустить» через НС и запомнить ответ, который выдает НС. Для объектов ВВ должна получиться таблица:

**Формулы для точности НС существенно зависит от типа задачи:**

* задача регрессии,
* задача классификации.

В любом случае все объекты ВВ нужно «пропустить» через НС и запомнить ответ, который выдает НС. Для объектов ВВ должна получиться таблица:

|  |  |  |
| --- | --- | --- |
| Объекты | Признак Y | Предсказанное значение |
| А1 | *y1* | *z1* |
| А2 | *y2* | *z2* |
| ... | ... | ... |
| At | *yt* | *zt* |

Наиболее популярная метрика качества в задаче регрессии



Прежде всего, информацию нужно занести в матрицу ошибок.

|  |  |  |  |
| --- | --- | --- | --- |
| Матрица ошибок | | Истинный класс | |
| 0 | 1 |
| Предсказанный класс | 0 | ? | ? |
| 1 | ? | ? |

Точность задачи классификации будет вычисляться по ячейкам матрицы ошибок.

Ячейки матрицы ошибок имеют собственные названия.

|  |  |  |  |
| --- | --- | --- | --- |
| Матрица ошибок | | Истинный класс | |
| 0 | 1 |
| Предсказанный класс | 0 | TN | FN |
| 1 | FP | TP |

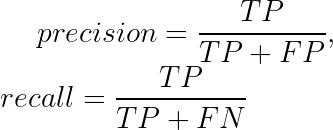
TN – true negative

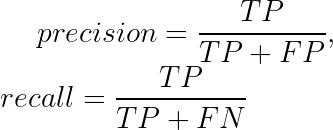
TP – true positive

FN – false negative

FP – false positive

**Точность и полнота (precision and recall)**





Это точность и полнота для класса 1.

Точность и полнота для класса 0 считаются аналогично.

Итак, качество классификации проверяется по двум характеристикам: **точность** и **полнота**.

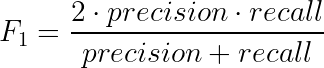
Это может приводить к появлению **несравнимых моделей**.

Например, 1-ая НС, точность 0.9 полнота 0.8

2-ая НС, точность 0.8 полнота 0.9

**Как выбрать лучшую?**

* мнение эксперта, какой параметр более важен для конкретной задачи;
* подсчитать агрегированную характеристику качества классификации.

**F1-значение (F1-value).** F1-значение позволяет вместо точности и полноты использовать только один параметр качества классификации.

Чем ближе *F1* к 1, тем лучше.

В задачах, в которых точность и полнота не равноценны применяют

(\beta>0)

Параметр (\beta>0)показывает **значимость полноты по сравнению с точностью.**

Если (\beta>0) =1, то мы получаем «обычное» значение *F1*.

# Выбор оптимальных гиперпараметров

У нейросети и в алгоритмах её тренировки есть величины (параметры), которые сами не настраиваются, а задаются до процесса обучения человеком.

У самой НС и в алгоритмах её тренировки есть величины (параметры), которые сами не настраиваются, а задаются до процесса обучения человеком.

Например:

* число слоев в НС, число нейронов в слое, итп (в общем, архитектура сети);
* шаг ГС *h*, константа *С* в регуляризации, итп (параметры обучения).

Такие величины называются **гиперпараметрами**.

**Главная проблема:** как задать оптимальное значение гиперпараметра?

По сути с помощью перебора.

1. Формируем список возможных значений параметра *B={b1,...,br}.*
2. Для каждого значения *bi* случайным образом делим РВ на ТВi и ВВi.
3. Тренируем НСi  на ТВi и считаем её точность на ВВi
4. Выбираем такое значение параметра *bi ,* на котором достигается максимальная точность.

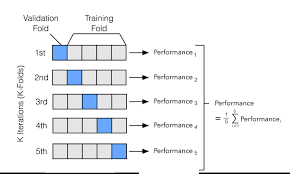
**Недостатки**: на точность параметра *bi*  существенно влияет конкретное разбиение на ТВ и ВВ.

Значит, каждое значение параметра *bi*  нужно проверять на нескольких разбиениях ТВ и ВВ.

**Кросс-валидация.** Разбиваем РВ на K фолдов (частей).

Каждое значение параметра *bi* участвует в построении K штук НС: НСi1,...,НСiK, причём каждый фолд ровно один раз является валидационным.

Для каждой НСij считаем точность.

Итоговая точность для значения параметра *bi* - это среднее арифметическое точностей K нейронных сетей НСi1,...,НСiK.

РВ в идеале нужно разбить на три части: ТВ, ВВ и тестовую (ТеВ) выборки (стандартная пропорция 70% 10% 20%)

ТеВ в процессе тренировки и выбора оптимальных гиперпараметров вообще не участвует.

Когда будут выбраны оптимальные параметры, то итоговая НС проверяется на ТеВ. Полученная точность считается окончательной – ну а дальше решение за заказчиком.

После каждой эпохи обучения вычислять точность НС на ВВ. Если ошибка на ВВ перестала уменьшаться, то уменьшить шаг h. Можно вообще остановить обучение, когда ошибка на ВВ начала расти.

# Переобучение и недообучение

Если ошибка НС даже на ТВ высокая, то налицо **недообучение (underfitting)**.

**Что делать**: нужно взять для тренировки более мощную нейросеть (с большим числом нейронов).

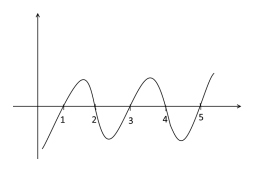
Если ошибка НС на ТВ низкая, но ошибка на ВВ высокая, то возникает **переобучение (overfitting)**.

В чем причина?

* + НС просто запомнила всю ТВ (это как студент, который просто запомнил примеры из учебника, а закономерность не понял);
  + НС реализовала неправильную закономерность.

НС восстановила закономерность:

|  |  |
| --- | --- |
| Номер курса, X | Самооценка студента, Y |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |



А если ВВ состоит из «промежуточных» точек x=1.5, 2.5, 3.5, то ошибка на ВВ огроменная!

**Что делать?**

* регуляризация,
* дропаут,
* больше данных.