Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных систем и сетей
Кафедра Информатики

Реферат
на тему:

Разделение выборки на обучающую, валидационную и тестовую. Баланс смещения и разброса. Кросс-валидация.

Студент М. С. Петрусевич
Проверил М. В. Стержанов

Минск 2020

Содержание

1 Разделение выборки на обучающую, валидационную и тестовую.

Одним из важных этапов разработки нейронных сетей с учителем является разделение выборки данных. Дело в том, что изначально, пришедшие данные, на которых будет обучаться нейронная сеть, представляют собой довольной сырой набор данных, которые не позволяют в полной мере сказать, что именно эти данных из себя представляют. Важными этапами являются такие шаги как разметка этих данных - т.е. определение, к какому классу объектов относится определённый набор данных и последующее разделение этих данных на выборки.

Если в случае с разметкой классов всё в большей степени понятно, то встаёт вопрос о том, какую цель преследует разбиение данных на выборки? Ответ на данный вопрос будет дан далее, что позволит оценить преимущество подхода разделения входных данных на выборки.

В машинном обучении общей задачей является изучение и построение алгоритмов, которые могут учиться и делать прогнозы на основе данных. Такие алгоритмы работают, делая управляемые данными прогнозы или решения, путем построения математической модели из входных данных.

Данные, используемые для построения окончательной модели, обычно поступают из нескольких наборов данных. В частности, три набора данных обычно используются на разных этапах создания модели.

1.1 Обучающая выборка

Модель изначально помещается в обучающий набор данных, который представляет собой набор примеров, используемых для соответствия параметрам модели. Модель (например, нейронная сеть) обучается на наборе обучающих данных с использованием контролируемого метода обучения (например, градиентного спуска или стохастического градиентного спуска). На практике обучающий набор данных часто состоит из пар входного вектора (или скаляра) и соответствующего выходного вектора (или скаляра), который обычно обозначается как цель (или метка). [1].

Текущая модель запускается с набором обучающих данных и выдает результат, который затем сравнивается с целью, для каждого входного вектора в наборе обучающих данных. На основании результатов сравнения и используемого алгоритма обучения параметры модели корректируются. Подгонка модели может включать как выбор переменных, так и оценку параметров.

1.2 Валидационная выборка

Затем подобранная модель используется для прогнозирования ответов на наблюдения во втором наборе данных, называемом набором проверочных данных. Набор данных проверки обеспечивает беспристрастную оценку соответствия модели учебному набору данных при настройке гиперпараметров модели (например, количество скрытых единиц в нейронной сети). Наборы данных проверки могут быть использованы для регуляризации путем ранней остановки: прекратите обучение, когда ошибка в наборе данных проверки увеличивается, так как это является признаком переобучения. Эта простая процедура на практике усложняется тем фактом, что ошибка набора данных может колебаться во время обучения, создавая несколько локальных минимумов.

1.3 Тестовая выборка

Наконец, тестовый набор данных представляет собой набор данных, используемый для обеспечения объективной оценки окончательного соответствия модели учебному набору данных. Если данные в наборе тестовых данных никогда не использовались при обучении (например, при кросс-валидации), набор тестовых данных также называется "holdout dataset".

1.4 Разделение на выборки

Далее рассмотрим пример разбиения на выборки, для последующего их использования:

- 70% train, 15% val, 15% test;
- 80% train, 10% val, 10% test;
- 60% train, 20% val, 20% test.

Как показано на рисунке, давайте представим, что у нас есть три модели для рассмотрения: модель A, модель B и модель C. Это могут быть модели с разными архитектурами, или они могут быть разными вариантами одной модели. Применим к каждой модели следующие шаги:

- Случайная инициализация модели;
- Тренировка модели на обучающей выборке;
- Оценка модели на валидационной выборке;
- Выбор наилучшей модели на основе результатов;
- Оценка модели на тестовой выборке.

Почему же нельзя просто использовать один набор данных? Давайте представим, что произойдет, если использовать все данные в качестве обу-

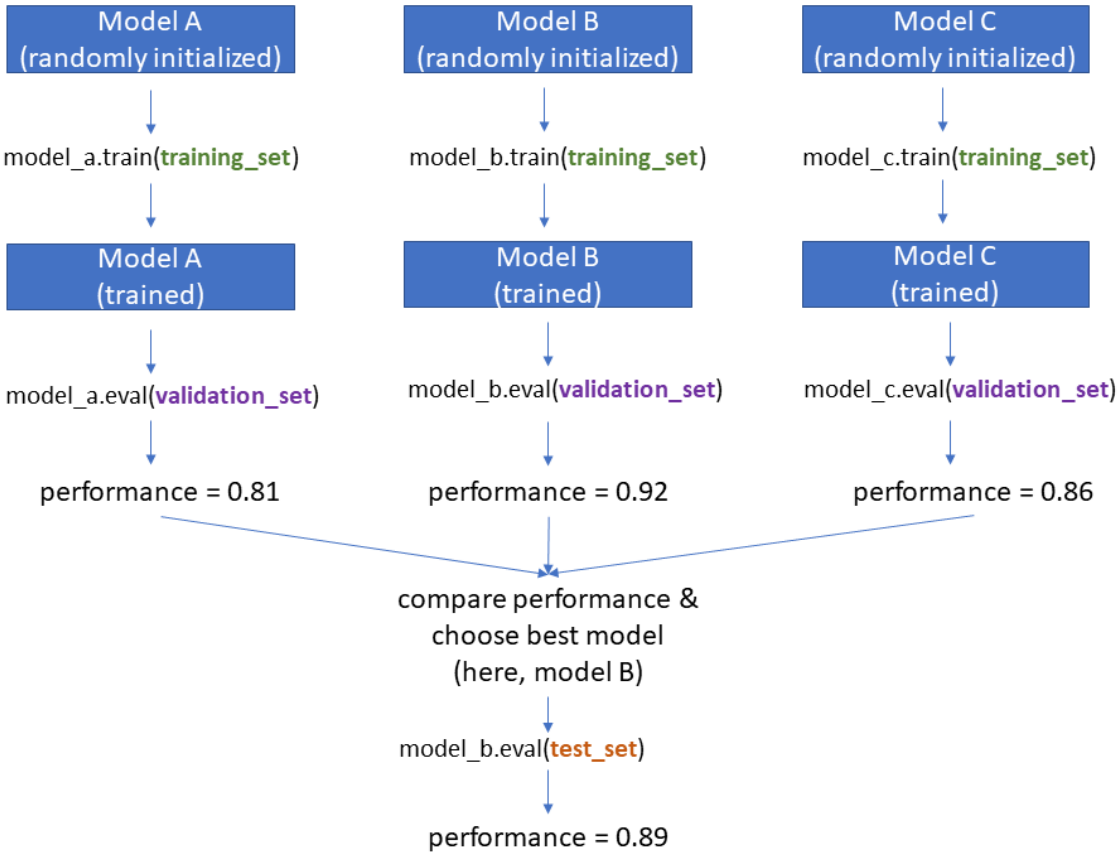

Рисунок 1.1 – Использование делений на выборки

чающей выборки. В случае оценки работы модели, оценивалась бы только работа модели на обучающей выборке. Теперь производительность обучающей выборки может отражать то, как модель будет работать с данными, которых она никогда не видела раньше. Но в плохом случае, модель просто запомнит примеры обучающих данных, и когда мы подадим ей пример, которого она никогда не видела, произойдёт неудача, т.к. модель переобучилась ("overfitting").

В данном случае нет возможности выяснить, везет ли нам или нет, и именно поэтому необходим набор для проверки работы модели. Валидационная выборка состоит из примеров, которые модель никогда не видела при обучении, поэтому, в случае хорошего результата на валидационной выборке, можно быть более уверенным, что модель получила полезные обобщаемые принципы.

Однако встаёт вопрос - для чего необходимо наличие и валидационной и тестовой выборки одновременно? Тестовая выборка важна, потому что во время выбора наилучшей модели может опять же возникнуть "overfitting". Рассмотрим это так: допустим, были опробованы тысячи различных моделей или вариантов моделей, и были найдено оптимальные па-

раметры для всех из них. Выбор модели с наилучшими характеристиками на валидационной выборке по своей сути означает что человек, выбравший определённую модель, настроил её на определённые значения. Точность модели, которая будет получена, по сути, будет завышена. Чтобы получить более точную и надёжную оценку того, насколько хороша эта "лучшая модель"будет работать с данными, которых она никогда не видела прежде, необходимо использовать больше данных, которых она никогда не видела раньше. Это и есть тестовая выборка. Точность на тестовой выборке, как правило, будет немного ниже, чем точность на валидационной выборке. [2]

Тогда встаёт вопрос о том, в каком соотношении нужно разделять выборки. Т.к. в идеале, хорошо бы иметь достаточное количество данных для каждой выборки, рассмотрим выгоду от каждой:

– Больше данных в обучающей выборке будет положительно влиять на конечный результат, т.к. модель сможет найти лучшее решение для подаваемых на вход значений, следовательно - лучше решает поставленную задачу. Если же обучающая выборка имеет небольшой размер,модель не сможет нормально обучиться и следовательно, будет иметь плохую точность.

– Больше данных для валидационной выборки так же влияет положительно, потому что это помогает принять лучшее решение о том, какая модель является "лучшей".

– Больше данных для тестовой выборки так же будет положительно сказываться, потому что это даст лучшее представление о том, насколько хорошо модель работает на данных, которые она раньше не видела.

В общем виде, можно сказать, что деление на выборки должно происходить в зависимости от, например, количества данных, которые имеются в распоряжении. Например, на небольших наборах данных, можно использовать некоторые ранее рассмотренные схемы - 70-15-15, 80-10-10, 60-20-20 и т.д. Основным принципом является то, что большая часть данных должна приходится на обучающую выборку. Однако, стоит отметить и подход, применяемый для больших наборов данных, так, например, есть смысл использовать схему 98-1-1 на достаточно больших наборах данных, т.к. даже 1 процента для валидационной и 1 процента для тестовой может вполне хватить, для настройки модели и проверки её точности. Так же, вполне возможно схемы вроде 98-1,5-0,5 и 99-0,5-0,5, но только в случаях больших наборов данных. Данный выбор схемы, обычно, ложится на плечи инженера, создающего модели и конкретные ситуации.

2 Баланс разброса и смещения

В моделях обучения с учителем алгоритм обучается на основе известных помеченных входных данных.

Целью любого обучения с учителем является наилучшая оценка функции (f) для выходной переменной (Y) с учетом входных данных (X). Функция-решение часто называется целевой функцией, потому что это та функция, которую данная модель машинного обучения стремится аппроксимировать.

Ошибка предсказания для любого алгоритма машинного обучения может быть разбита на две части:

– bias error;
– variance error.

2.1 Bias error

Bias - это упрощающие предположения, сделанные моделью для облегчения изучения целевой функции.

В общем случае, линейные алгоритмы имеют большое значение bias, что делает их быстрыми для изучения и более простыми для понимания, но в целом менее гибкими. В свою очередь, они имеют более низкую прогнозирующую производительность по сложным задачам.

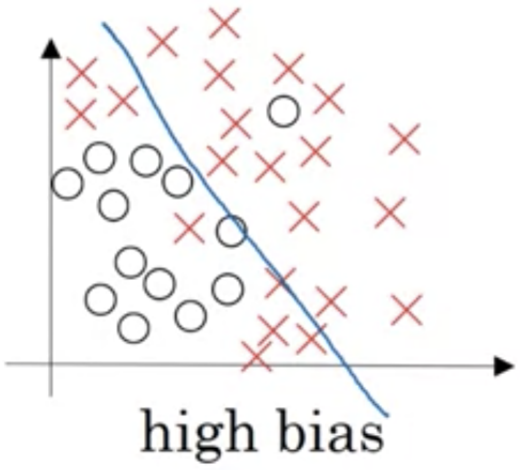

Рисунок 2.1 – Пример с high bias

2.2 Variance error

Variance - это величина, на которую изменится оценка целевой функции, если использовались разные обучающие данные.

Целевая функция оценивается по данным обучения с помощью алгоритма машинного обучения, поэтому следует ожидать, что алгоритм будет иметь некоторую дисперсию. В идеале, он не должен слишком сильно меняться от одного обучающего набора данных к следующему, что означает, что алгоритм хорош в выделении скрытого базового отображения между входными и выходными переменными.

Алгоритмы машинного обучения, которые имеют высокую дисперсию, сильно зависят от специфики данных обучения. Это означает, что специфика обучения влияет на количество и типы параметров, используемых для характеристики функции отображения. [3]

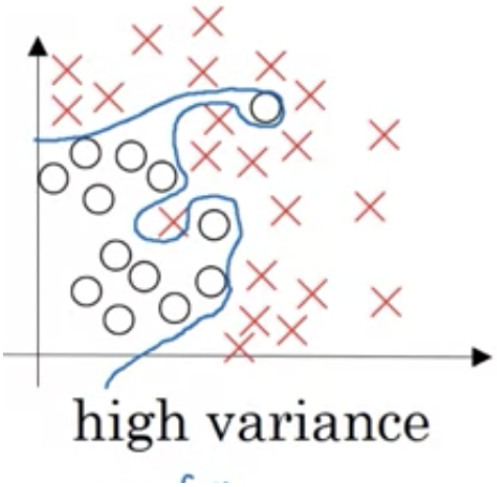

Рисунок 2.2 – Пример с high variance

2.3 Баланс bias и variance

Целью любого обучения с учителем является достижение низких показателей bias и variance. Также, в свою очередь, алгоритм должен обеспечивать необходимую точность прогнозирования с минимальными значениями разброса и смещения. Таким образом возникает задача нахождения оптимальных параметров, которые будут удовлетворять всем описанным выше критериям.

Можно отметить что:

– линейный алгоритмы машинного обучения чаще имеют high bias, но в то же время low variance;

– нелинейные алгоритмы машинного обучения же чаще наоборот - low bias и high variance.

Идеальным сочетанием будет является нахождения точки обучения как на следующем рисунке:

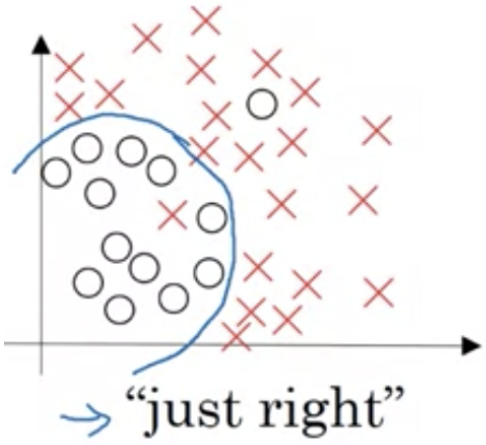

Рисунок 2.3 – Пример "идеально"обученной модели

Также можно отметить, что предположения о наличии high bias или high variance можно делать исходя из значения ошибок на выборках. К примеру, значения ошибки на обучающей выборке в 1% и 11% на валидационной, может сказать, что это проявление переобучения модели, т.е. high variance. в то же время значения 15% и 16% на обучающей и валидационной выборках, соответсвенно, говорят о high bias. Также, может быть случай одновременного наличия high bias и high variance (например 15% и 30% на обучающей и валидационной выборках соответсвенно). [4]

Для решения проблем с high bias и high variance существуют следующие способы, каждый из них зависит от того, какую именно проблему мы решаем. Рассмотрим проблему high bias. В данном случае решением будут:

– увеличить размерность сети;
– увеличить время настройки модели.

В случае с high variance, для того, чтобы победить переобучение можно использовать:

– добавить больше данных;
– добавить регуляризацию (или перенастроить её, если она уже есть).

Применяя данные техники можно постараться уменьшить негативное влияние high bias и high variance на модель.

3 Кросс-валидация

Кросс-валидация - это статистический метод, используемый для оценки моделей машинного обучения.

Он обычно используется в прикладном машинном обучении для сравнения и выбора модели для данной задачи прогнозного моделирования, потому что это легко понять, легко реализовать и приводит к оценкам, которые, как правило, имеют меньшую систематическую ошибку, чем другие методы.

Процедура имеет единственный параметр, называемый k, который относится к числу групп, на которые следует разбить данную выборку данных. Таким образом, процедуру часто называют перекрестной проверкой в k-кратном порядке. Когда выбрано конкретное значение для k, оно может использоваться вместо k в ссылке на модель, например, k = 10 становится 10-кратной кросс-валидацией.

Кросс-валидация в основном используется в прикладном машинном обучении для оценки навыка модели машинного обучения на невидимых данных. То есть использовать ограниченную выборку, чтобы оценить, как ожидается, что модель в целом будет работать, когда она используется для прогнозирования данных, которые не использовались во время обучения модели.

Это популярный метод, потому что его легко понять и потому что он обычно дает менее предвзятую или менее оптимистичную оценку навыка модели, чем другие методы, такие как простое разделение на выборки. [5]

Кросс-валидация представлена следующими шагами:

– перемешать датасет;

– разделить датасет на k групп;

– для каждой группы необходимо взять её как тестовую выборку, остальные - как обучающую и натренировать модель исходя из этого.

– подвести итог работы модели.

Важно отметить, что каждое наблюдение в выборке данных присваивается отдельной группе и остается в этой группе в течение всей процедуры. Это означает, что каждой "итерации"дается возможность быть использованной в удерживающем наборе 1 раз и использоваться для обучения модели k-1 раз.

Также важно, чтобы любая подготовка данных до подгонки модели происходила из набора обучающих данных, назначенного в процедуре кросс-валидации, в цикле, а не в более широком наборе данных. Это также относится к любой настройке гиперпараметров. Невыполнение этих опера-

ций в цикле может привести к утечке данных и оптимистической оценке навыка модели.

Результаты использования кросс-валидации в k-кратном порядке часто суммируются со средним значением баллов по модели. Хорошей практикой также является включение показателя дисперсии оценок навыков, таких как стандартное отклонение или стандартная ошибка.

4 Вывод

В данном реферате были рассмотрены техники разбиения датасетов на выборки, разрешения проблем баланса смещения и разброса, а также техники кросс-валидации. Использование множества данных техник и подходов позволяет более удачно использовать модели машинного обучения и настраивать их, вырабатывая из них максимум.

Были рассмотрены разные разбиения выборок на обучающую, валидационную и тестовую. Можно отметить, что выбор того, в каком процентом соотношении нужно делить выборки лежит всецело на разработчике, т.к. обобщённые подходы используемые в индустрии лишь примерно показывают в каком соотношении лучше делить выборки, т.е. они являются рекомендательными и не универсалины для всех тип задач. Правильный подбор соотношения train-validation-test выборок позволит вовремя обнаружить проблемы bias'а или variance'а, что позволят сократить время, которое будет затрачено на построение и отладку модели, а также повысит её конечную точность.

Для того, чтобы не попадать в ситуации переобучения используются различные техники, такие как l1-регуляризация, l2-регуляризация или использования Dropout слоя в нейронной сети. Случаи с недообученной нейронной сетью могут решаться добавлением большего количества данных или увеличением размера обучающей выборки (например в случае, когда под валидационную выборку было взято слишком много данных не имея при этом веской причины).

Описанные выше техники позволяют позволят обучать нейронные сети с более высокой скоростью за счёт того, что на основе анализа bias'ов и variance'ов можно делать выводы о том, в какую сторону изменять гиперпараметры модели, хватает ли обучающих данных для данной задачи и не происходило ли каких-либо аномалий при обучении модели.

Список использованных источников

[1] racheldraelos. Best Use of Train/Val/Test Splits, with Tips for Medical Data. — 2019. https://glassboxmedicine.com/2019/09/15/best-use-of-train-val-test-splits-with-tips-for-medical-data/.

[2] Brownlee, Jason. What is the Difference Between Test and Validation Datasets? — 2018. https://machinelearningmastery.com/difference-test-validation-datasets/.

[3] Brownlee, Jason. Gentle Introduction to the Bias-Variance. — 2018. https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning/.

[4] Ng, Andrew. Basic Recipe for Machine Learning. — 2016. https://www.coursera.org/learn/deep-neural-network/lecture/ZBkx4/basic-recipe-for-machine-learning.

[5] Brownlee, Jason. A Gentle Introduction to k-fold Cross-Validation. — 2018. https://machinelearningmastery.com/k-fold-cross-validation/.