Google News Initiative

LESSON 5

How does a machine learn?

A step-by-step overview of the ML training process.

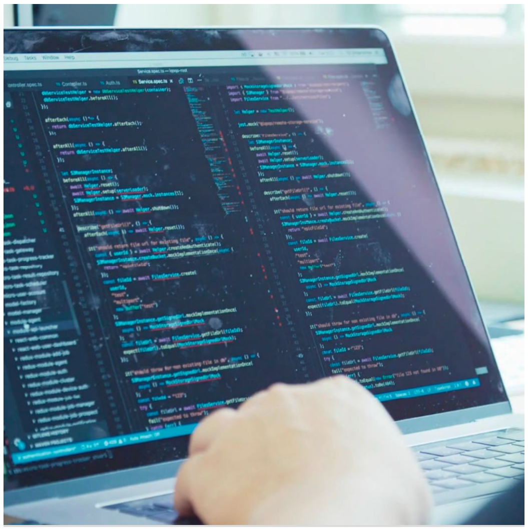
Lesson Overview

Training your Machine Learning model

So far, we have hinted at the fact that a ML model needs to be 'trained' in order to produce the expected outcome. In this lesson you will learn what steps are involved in the training process, through the lens of a specific case-study.

The goal is to help you understand how machines learn, not yet to be able to replicate the process on your own.

Before you decide to use machine learning, ask yourself: What question am I trying to find answers for? And do I need machine learning to get there?



1 What question do you want to answer?

2 Assessing your use case

3 Getting the data

4 Getting the data in shape

5 Choosing an algorithm

6 Training, validating and testing the model

7 Evaluating the results

8 Journalistic evaluation

For more lessons, visit:

newsinitiative.withgoogle.com/training

What question do you want to answer?

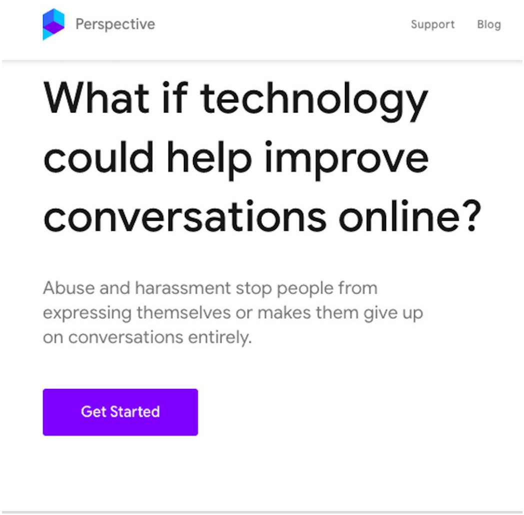
SINGLE STEP

Imagine your website gives readers the opportunity to comment on articles. Every day thousands of comments are posted and, as it happens, sometimes the conversation gets a little nasty.

It would be great if an automated system could categorise all comments posted on your platform, identify those that might be 'toxic' and flag them to the human moderators, who could review them to improve the quality of the conversation.

That's a type of problem machine learning can help you with. And in fact, it already does. Check Jigsaw's Perspective API to find out more.

This is the example we are going to use to learn how a machine learning model is trained.But keep in mind that the same process can be extended to any number of different case-studies.



What if technology could help improve conversations online?

Abuse and harassment stop people from expressing themselves or makes them give up on conversations entirely.

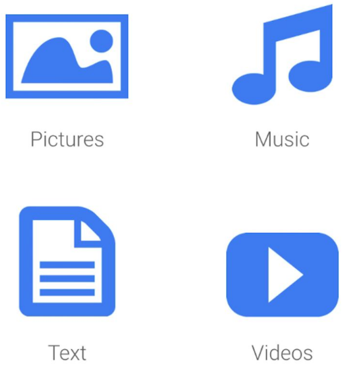
Assessing your use case

SINGLE STEP

To train a model to recognise toxic comments, you need data. Which in this case means examples of comments you receive on your website. But before you prepare your dataset, it’s important to reflect on what is the outcome you are trying to achieve.

Even for humans, it's not always easy to evaluate whether a comment is toxic and should therefore not be published online. Two moderators might have different views on the ‘toxicity’ of a comment. So you shouldn't expect the algorithm to magically "get it right" all the time.

Machine learning can handle a huge number of comments in minutes, but it's important to keep in mind that it's just ‘guessing’ based on what it learns. It will sometimes give wrong answers and generally, make mistakes.



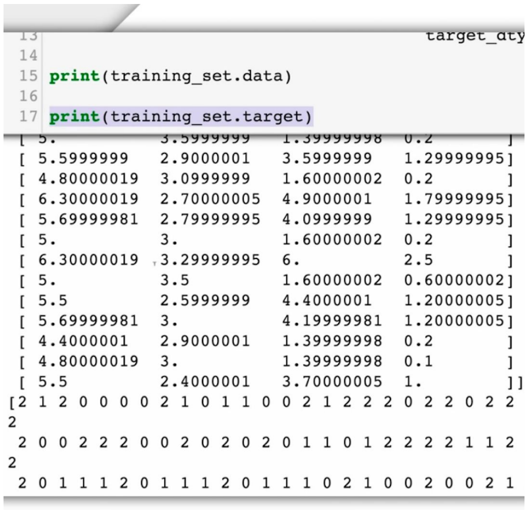
Getting the data

SINGLE STEP

It's now time to prepare your dataset. For our case-study, we already know what kind of data we need and where to find it: comments posted on your website.

Since you are asking the machine learning model to recognise comments' toxicity, you need to supply labelled examples of the kinds of text items you want to classify (comments), and the categories or labels you want the ML system to predict ("toxic" or "non toxic").

For other use cases you might not have the data so easily available, though. You will need to source it from what your organisation collects or from third-parties. In both cases, make sure to review regulations about data protection in both your region and the locations your application will serve.



```
                                                  target_dty

print(training_set.data)

print(training_set.target)
```

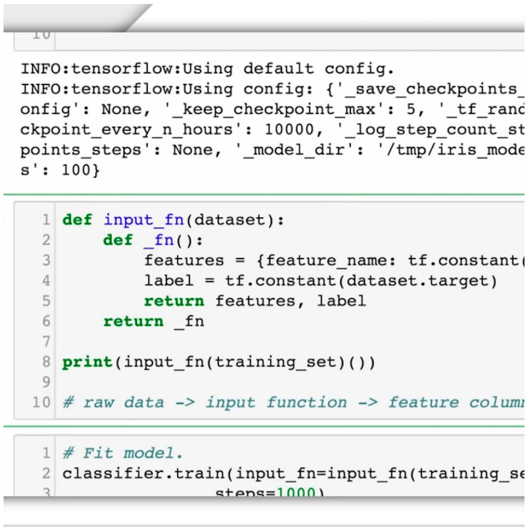
```
 [ 5.          3.5999999   1.39999998  0.2       ]
 [ 5.5999999   2.9000001   3.5999999   1.29999995]
 [ 4.80000019  3.0999999   1.60000002  0.2       ]
 [ 6.30000019  2.70000005  4.9000001   1.79999995]
 [ 5.69999981  2.79999995  4.0999999   1.29999995]
 [ 5.          3.          1.60000002  0.2       ]
 [ 6.30000019  3.29999995  6.          2.5       ]
 [ 5.          3.5         1.60000002  0.60000002]
 [ 5.5         2.5999999   4.4000001   1.20000005]
 [ 5.69999981  3.          4.19999981  1.20000005]
 [ 4.4000001   2.9000001   1.39999998  0.2       ]
 [ 4.80000019  3.          1.39999998  0.1       ]
 [ 5.5         2.4000001   3.70000005  1.        ]]
[2 1 2 0 0 0 0 2 1 0 1 1 0 0 2 1 2 2 2 0 2 2 0 2 2
2
 2 0 0 2 2 2 0 0 2 0 2 0 2 0 1 1 0 1 2 2 2 2 1 1 2
2
 2 0 1 1 1 2 0 1 1 1 2 0 1 1 1 0 2 1 0 0 2 0 0 2 1
```

Getting your data in shape

SINGLE STEP

Once you have collected the data and before you feed it to the machine, you need to analyse the data in depth. The output of your machine learning model will be only as good and fair as your data is (more on the concept of 'fairness' in the next lesson). You must reflect on how your use case might negatively impact the people that will be affected by the actions suggested by the model.

Among other things, in order to successfully train the model you will need to make sure to include enough labelled examples and to distribute them equally across categories. You must also provide a broad set of examples, considering the context and the language used, so that the model can capture the variation in your problem space.



```
INFO:tensorflow:Using default config.
INFO:tensorflow:Using config: {'_save_checkpoints_
onfig': None, '_keep_checkpoint_max': 5, '_tf_rand
ckpoint_every_n_hours': 10000, '_log_step_count_st
points_steps': None, '_model_dir': '/tmp/iris_mode
s': 100}
```

```
def input_fn(dataset):
    def _fn():
        features = {feature_name: tf.constant(
        label = tf.constant(dataset.target)
        return features, label
    return _fn

print(input_fn(training_set)())

# raw data -> input function -> feature colum
```

```
# Fit model.
classifier.train(input_fn=input_fn(training_se
                 steps=1000)
```

Choosing an algorithm

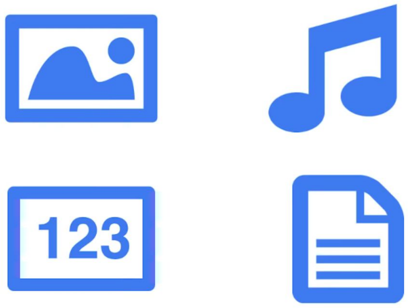
SINGLE STEP

After you are done preparing the dataset, you have to choose a machine learning algorithm to train. Every algorithm has its own purpose. Consequently, you must pick the right kind of algorithm based on the outcome you want to achieve.

In previous lessons we have learned about different approaches to machine learning. Since our case-study requires labelled data in order to be able to classify our comments as "toxic" or "non toxic", what we are trying to do is supervised learning.

Google Cloud AutoML Natural Language is one of many algorithms that allow you to achieve our desired outcome. But whatever algorithm you choose, make sure to follow the specific instructions on how it requires the training dataset to be formatted.

Choosing a Model



Training, validating and testing the model

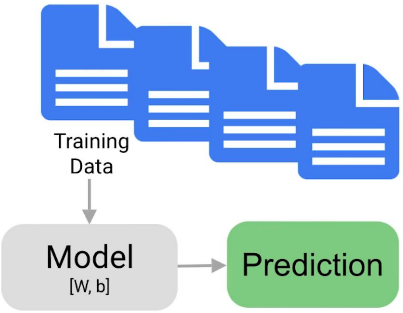
SINGLE STEP

Now we move onto what is the proper training phase, in which we use the data to incrementally improve our model's ability to predict if a given comment is toxic or not. We feed most of our data to the algorithm, perhaps wait a few minutes, and voilà, our model is trained.

But why only "most" of the data? To make sure the model learns properly, you must divide your data in three:

- The training set is what your model "sees" and initially learns from.
- The validation set is also part of the training process but it's kept separate to tune the model's hyperparameters, variables that specify the model's structure.
- The test set enters the stage only after the training process. We use it to test the performance of our model on data it has not yet seen.

Parameter Tuning



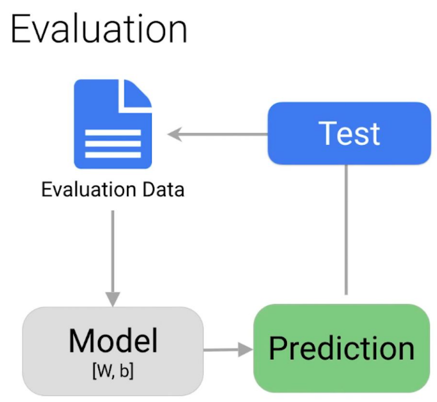
Evaluating the results

SINGLE STEP

How do you know if the model has correctly learned to spot potentially toxic comments?

When the training is complete, the algorithm provides you with an overview of the model performance. As we already discussed, you can't expect the model to get it right 100% of the time. It's up to you to decide what is ‘good enough’ depending on the situation.

The main things you want to consider to evaluate your model are false positives and false negatives. In our case, a false positive would be a comment that is not toxic, but gets marked as such. You can quickly dismiss it and move on. A false negative would be a comment that is toxic, but the system fails to flag it as such. It's easy to understand which mistake you should want your model to avoid.



Journalistic evaluation

SINGLE STEP

Evaluating the results of the training process doesn't end with the technical analysis. At this point, your journalistic values and guidelines should help you decide if and how to use the information the algorithm is providing.

Start by thinking whether you now have information that was not available before, and about the newsworthiness of that information. Does it validate your existing hypothesis or is it shedding light on new perspectives and story angles you were not considering before?

You should now have a better understanding of how machine learning works and you might be even more curious to try out its potential. But we are not ready yet. The next lesson will introduce the number one concern machine learning brings with it: Bias.