Get started Open in app

towards
data science

Follow 593K Followers

Automating Machine Learning Pipelines

Using Model Search for Automating Machine Learning

Himanshu Sharma 1 day ago · 3 min read★

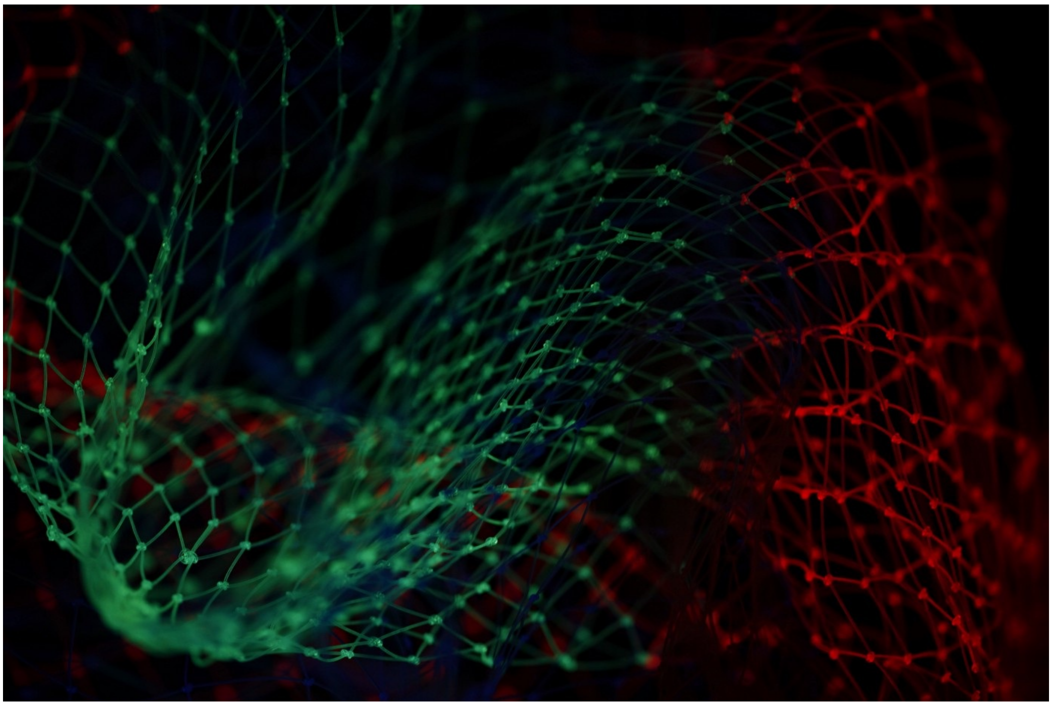


Photo by Pietro Jeng on Unsplash

Creating a Machine Learning model is a difficult task because we need to write a lot of code to try different models and find out the performing model for that particular problem. There are different libraries that can automate this process to find out the best performing Machine Learning model but they also require some coding.

What if I tell you that we can run multiple AutoML algorithms to find out the best model architecture for classification problems in a single code cell? Model search helps in implementing AutoML for classification problems. It runs multiple ML algorithms and compares them with each other. It creates a search space where we can customize the type of layer we want in the neural network.

Model Search, an open-source Python library that offers an effortless way to run multiple AutoML and find out the best one. It is highly useful time-saving and works effortlessly.

In this article, we will explore Model Search and some of its functionalities.

Let’s get started…

Installing required libraries

We will start by installing Model Search using pip installation. The command given below will install Model Search using pip.

```
!git clone https://github.com/google/model_search.git
%cd /content/model_search/
```

Importing required libraries

In this step, we will import all the libraries that are required for creating the model and train the models to find the best model.

```
import model_search
from model_search import constants
from model_search import single_trainer
from model_search.data import csv_data
```

Creating the Model

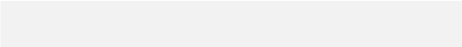
Now we will directly create the model. We will use the CSV data already stored in the model search. In order to create the model we just pass all the required parameters and start training.

```
trainer = single_trainer.SingleTrainer(
    data=csv_data.Provider(
        label_index=0,
        logits_dimension=2,
        record_defaults=[0, 0, 0, 0],
        filename="model_search/data/testdata/csv_random_data.csv"),

spec='/content/model_search/model_search/configs/dnn_config.pbtxt')

trainer.try_models(
    number_models=200,
    train_steps=1000,
    eval_steps=100,
    root_dir="/tmp/run_example",
    batch_size=32,
    experiment_name="example",
    experiment_owner="model_search_user")
```

Model Training(Source: By Author)

As we can see here the models are being trained. Now we can also print the best model and its parameters.

```
!ls /tmp/run_example/tuner-1/
```



Output(Source: By Author)

Here you can see the best-performing model. All the model files are saved in your local machine in the run_example folder. Go ahead try this with different datasets and create different models and let me know your comments in the response section.

This article is in collaboration with Piyush Ingale.

Before You Go

Thanks *for reading! If you want to get in touch with me, feel free to reach me at hmix13@gmail.com or my* ***LinkedIn Profile****. You can view my* ***Github*** *profile for different data science projects and packages tutorials. Also, feel free to explore* ***my profile*** *and read different articles I have written related to Data Science.*

Data Science · Data Visualization · Python · Machine Learning · Artificial Intelligence