

Getting started with GitHub Codespaces for machine learning

In this article

Introduction

Building a simple image classifier

Opening your codespace in JupyterLab

Configuring NVIDIA CUDA for your codespace

Learn about working on machine learning projects with GitHub Codespaces and its out-of-the-box tools.

Introduction

This guide introduces you to machine learning with GitHub Codespaces. You'll build a simple image classifier, learn about some of the tools that come preinstalled in GitHub Codespaces, and find out how to open your codespace in JupyterLab.

Building a simple image classifier

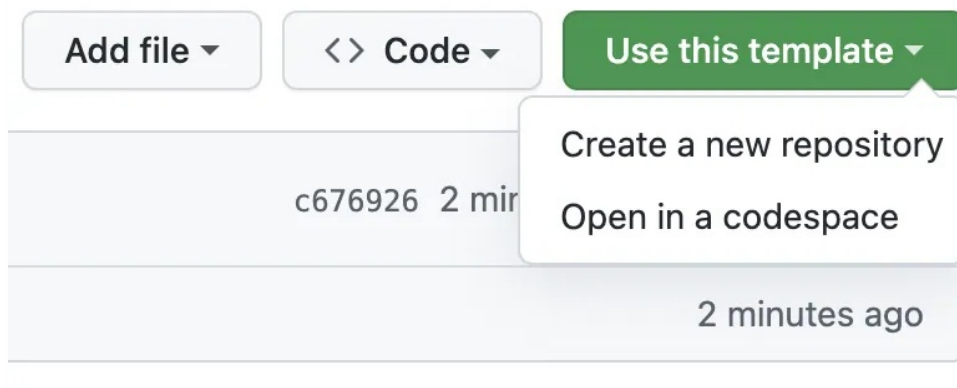
We'll use a Jupyter notebook to build a simple image classifier.

Jupyter notebooks are sets of cells that you can execute one after another. The notebook we'll use includes a number of cells that build an image classifier using [PyTorch](#). Each cell is a different phase of that process: download a dataset, set up a neural network, train a model, and then test that model.

We'll run all of the cells, in sequence, to perform all phases of building the image classifier. When we do this Jupyter saves the output back into the notebook so that you can examine the results.

Creating a codespace

- 1 Go to the [github/codespaces-jupyter](https://github.com/github/codespaces-jupyter) template repository.
- 2 Click **Use this template**, then click **Open in a codespace**.



A codespace for this template will open in a web-based version of Visual Studio Code.

Opening the image classifier notebook [🔗](#)

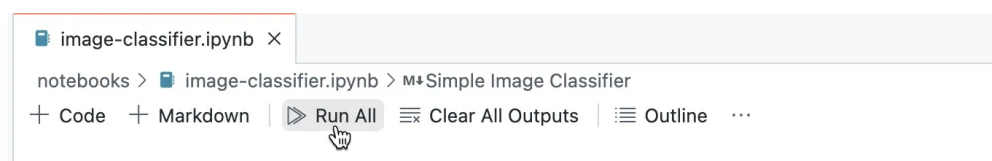
The default container image that's used by GitHub Codespaces includes a set of machine learning libraries that are preinstalled in your codespace. For example, Numpy, pandas, SciPy, Matplotlib, seaborn, scikit-learn, Keras, PyTorch, Requests, and Plotly. For more information about the default image, see "[Introduction to dev containers](#)" and the [devcontainers/images](#) repository].

- 1 In the VS Code editor, close any "Get Started" tabs that are displayed.
- 2 Open the `notebooks/image-classifier.ipynb` notebook file.

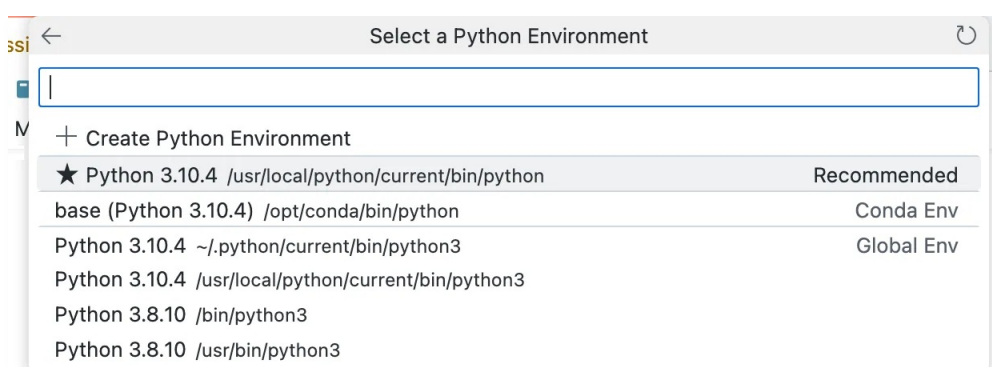
Building the image classifier [🔗](#)

The image classifier notebook contains all the code you need to download a dataset, train a neural network, and evaluate its performance.

- 1 Click **Run All** to execute all of the notebook's cells.



- 2 If you are prompted to choose a kernel source, select **Python Environments**, then select the version of Python at the recommended location.



- 3 Scroll down to view the output of each cell.

Step 3: Train the network and save model

PyTorch trains our network by adjusting its parameters and evaluating its performance against our labelled dataset.

```
from tqdm import tqdm

EPOCHS = 2
print("Training...")
for epoch in range(EPOCHS):
    running_loss = 0.0
    for i, data in enumerate(tqdm(trainloader, desc=f"Epoch {epoch + 1} of {EPOCHS}", leave=True, ncols=80)):
        inputs, labels = data

        optimizer.zero_grad()
        outputs = net(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

# Save our trained model
PATH = './cifar_net.pth'
torch.save(net.state_dict(), PATH)
```

[3] 29.6s Python

... Training...

Epoch 1 of 2: 100%|██████████| 5000/5000 [00:23<00:00, 213.55it/s]
Epoch 2 of 2: 26%|███████| 1322/5000 [00:06<00:16, 223.53it/s]

Opening your codespace in JupyterLab [↗](#)

You can open your codespace in JupyterLab from the "Your codespaces" page at github.com/codespaces, or by using GitHub CLI. For more information, see "[Opening an existing codespace](#)."

The JupyterLab application must be installed in the codespace you are opening. The default container image includes JupyterLab, so codespaces created from the default image will always have JupyterLab installed. For more information about the default image, see "[Introduction to dev containers](#)" and [the devcontainers/images repository](#). If you're not using the default image in your dev container configuration, you can install JupyterLab by adding the `ghcr.io/devcontainers/features/python` feature to your `devcontainer.json` file. You should include the option `"installJupyterlab": true`. For more information, see [the README for the python feature](#), in the `devcontainers/features` repository.

Configuring NVIDIA CUDA for your codespace [↗](#)

Note: This section only applies to customers who can create codespaces on machines that use a GPU. The ability to choose a machine type that uses a GPU was offered to selected customers during a trial period. This option is not generally available.

Some software requires you to install NVIDIA CUDA to use your codespace's GPU. Where this is the case, you can create your own custom configuration, by using a `devcontainer.json` file, and specify that CUDA should be installed. For more information on creating a custom configuration, see "[Introduction to dev containers](#)."

For full details of the script that's run when you add the `nvidia-cuda` feature, see the [devcontainers/features](#) repository.

- 1 Within the codespace, open the `.devcontainer/devcontainer.json` file in the editor.
- 2 Add a top-level `features` object with the following contents:

JSON



```
"features": {
  "ghcr.io/devcontainers/features/nvidia-cuda:1": {
```

```
    "installCudnn": true
  }
}
```

For more information about the `features` object, see the [development containers specification](#).

If you are using the `devcontainer.json` file from the image classifier repository you created for this tutorial, your `devcontainer.json` file will now look like this:

```
{
  "customizations": {
    "vscode": {
      "extensions": [
        "ms-python.python",
        "ms-toolsai.jupyter"
      ]
    }
  },
  "features": {
    "ghcr.io/devcontainers/features/nvidia-cuda:1": {
      "installCudnn": true
    }
  }
}
```

- 3 Save the change.
- 4 Access the VS Code Command Palette (`Shift + Command + P` / `Ctrl + Shift + P`), then start typing "rebuild". Click **Codespaces: Rebuild Container**.



Tip: You may occasionally want to perform a full rebuild to clear your cache and rebuild your container with fresh images. For more information, see "[Rebuilding the container in a codespace](#)."

The codespace container will be rebuilt. This will take several minutes. When the rebuild is complete the codespace is automatically reopened.

- 5 Publish your change to a repository so that CUDA will be installed in any new codespaces you create from this repository in future. For more information, see "[Creating a codespace from a template](#)."

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)