Introduction to Deep Learning
GitHub Codespaces - Backup Setup Guide

MSc Computer Science

Week 1

Contents

1 Introduction

Important

This guide is a **backup option** for students who are having trouble with local installation. You should still try to get your local environment working, as it's better for long-term development skills.

Note

GitHub Codespaces provides a cloud-based development environment with VS Code in your browser. It's free for students with 60 hours/month of usage.

1.1 When to Use Codespaces

Use GitHub Codespaces if:

- Your local Python installation is broken and you can't fix it quickly
- You're waiting for IT support to resolve installation issues
- You need to work from a different computer temporarily
- Your computer doesn't have enough disk space

1.2 Advantages

- No local installation needed
- Works from any computer with a web browser
- Pre-configured environment
- Integrated with GitHub for easy code management
- VS Code interface (similar to local development)
- Free GPU access (limited)

1.3 Limitations

- Requires internet connection
- 60 hours/month free quota (should be enough for this course)
- Session timeouts after 30 minutes of inactivity
- Files are saved in the cloud (make sure to commit/download)

2 Prerequisites

2.1 GitHub Account

You need a GitHub account:

1. Go to `https://github.com/signup`
2. Create a free account
3. Verify your email address

2.2 GitHub Student Benefits (Recommended)

Get free GitHub Pro with student benefits:

1. Go to `https://education.github.com/pack`
2. Click "Sign up for Student Developer Pack"
3. Verify your student status (using university email)
4. Get increased Codespaces quota and other benefits

3 Setting Up Your Repository

3.1 Option A: Fork Course Repository (Recommended)

If the instructor provides a course repository:

1. Go to the course repository URL (provided by instructor)
2. Click the "Fork" button (top right)
3. This creates your own copy of the repository
4. You now have your own repository: `github.com/YourUsername/deep-learning-course`

3.2 Option B: Create New Repository

If starting from scratch:

1. Go to `https://github.com/new`
2. Repository name: `deep-learning-course`
3. Description: "Deep Learning Course - Week 1"
4. Select "Public" or "Private"
5. Check "Add a README file"
6. Check "Add .gitignore" → select "Python"
7. Click "Create repository"

4 Creating a Codespace

4.1 From Your Repository

1. Navigate to your repository on GitHub
2. Click the green "Code" button
3. Select the "Codespaces" tab
4. Click "Create codespace on main"
5. Wait 1-2 minutes for the environment to build

Note

The first time you create a Codespace, it may take a few minutes to set up. Subsequent launches will be faster.

4.2 Codespace Interface

Once loaded, you'll see:

- Left sidebar: File explorer, Git, extensions
- Center: Code editor
- Bottom: Terminal
- Top: Menu bar

It looks and works like VS Code!

5 Setting Up Python Environment

5.1 Check Python Version

In the terminal (bottom panel), run:

```
python3 --version
```

Should show Python 3.10 or later.

5.2 Create Virtual Environment

```
# Create virtual environment
python3 -m venv deep_learning_env

# Activate it
source deep_learning_env/bin/activate

# You should see (deep_learning_env) in the prompt
```

5.3 Install Required Packages

5.3.1 Using requirements.txt

If you have the `requirements.txt` file:

1. Upload `requirements.txt` to your repository:
   - Drag and drop into file explorer, or
   - Use terminal: create file and paste content
2. Install packages:

```
pip install --upgrade pip
pip install -r requirements.txt
```

5.3.2 Manual Installation

```
pip install torch torchvision torchaudio
pip install jupyter notebook
pip install matplotlib numpy pandas
pip install ipywidgets
```

> **Tip**
>
> Installation in Codespaces is usually faster than local due to GitHub's fast network connection.

6 Using Jupyter Notebooks

6.1 Opening Notebooks

1. Upload your `.ipynb` file:
   - Drag and drop into file explorer
   - Or clone from another repository
2. Click on the `.ipynb` file to open it
3. VS Code will render it as a Jupyter notebook

6.2 Selecting Kernel

1. Click "Select Kernel" button (top right of notebook)
2. Choose "Python Environments"
3. Select your `deep_learning_env`

6.3 Running Cells

- Click the play button next to a cell, or
- Press Shift + Enter to run current cell and move to next
- Press Ctrl/Cmd + Enter to run current cell

7 Working with Files

7.1 Uploading Files

1. Drag and drop files into the file explorer (left sidebar)
2. Or use terminal:

```
# Download from URL
wget https://example.com/file.ipynb

# Or upload via GitHub web interface, then git pull
```

7.2 Downloading Files

1. Right-click file in explorer → Download
2. Or commit and push to GitHub, then download from there

7.3 Saving Your Work

Important

Codespaces can timeout! Make sure to save your work regularly.

Method 1: Git Commit (Recommended)

```
# Stage all changes
git add .

# Commit with message
git commit -m "Week 1 exercises completed"

# Push to GitHub
git push
```

Method 2: Auto-save

- Files are auto-saved in Codespace
- But will be lost if Codespace is deleted
- Always commit important work!

8 Managing Your Codespace

8.1 Stopping a Codespace

When you're done working:

1. Click "Codespaces" in bottom left corner
2. Select "Stop Current Codespace"
3. Or just close the browser tab (auto-stops after 30 min)

Tip

Stopping your Codespace saves your free quota hours!

8.2 Restarting a Codespace

1. Go to `https://github.com/codespaces`
2. Find your codespace in the list
3. Click the name to reopen it
4. Remember to reactivate virtual environment:

```
source deep_learning_env/bin/activate
```

8.3 Deleting a Codespace

If you want to start fresh:

1. Go to `https://github.com/codespaces`
2. Click "..." next to your codespace
3. Select "Delete"
4. Create a new one from your repository

9 Tips and Best Practices

9.1 Keyboard Shortcuts

Learn these VS Code shortcuts:

- Ctrl/Cmd + P: Quick file open
- Ctrl/Cmd + Shift + P: Command palette
- Ctrl/Cmd + `: Toggle terminal
- Ctrl/Cmd + B: Toggle sidebar

9.2 Extensions

Install helpful extensions:

1. Click Extensions icon (left sidebar)
2. Search and install:
    - Python (Microsoft) - should be pre-installed
    - Jupyter (Microsoft) - should be pre-installed
    - Pylance - better Python IntelliSense

9.3 Git Integration

Use the built-in Git interface:

1. Click Source Control icon (left sidebar)
2. See changed files
3. Stage changes (+ button)
4. Write commit message
5. Click checkmark to commit
6. Click "..." $\rightarrow$ Push

9.4 Terminal Tips

```
# Create new terminal: Ctrl+Shift+`

# Split terminal: Click split icon

# Multiple terminals for different tasks:
# Terminal 1: Jupyter notebook
# Terminal 2: General commands
```

10 Common Issues

10.1 Codespace Won't Start

- Check your internet connection
- Wait a few minutes and try again
- Check GitHub status: `https://www.githubstatus.com/`
- Try creating a new Codespace

10.2 Out of Free Hours

- You get 60 hours/month free (120 with Student Pack)
- Check usage: `https://github.com/settings/billing`
- Stop Codespaces when not in use
- Use local environment when possible

10.3 Kernel Died / Module Not Found

1. Make sure virtual environment is activated:

   ```
   source deep_learning_env/bin/activate
   ```

2. Reinstall packages if needed
3. Restart kernel: Click "Restart" button in notebook toolbar

10.4 Files Not Saving

1. Check file is not read-only
2. Manually save: Ctrl/Cmd + S
3. Check Codespace has not timed out
4. Commit to Git to be safe

11 Transitioning to Local Setup

Once your local installation is working:

11.1 Downloading Your Work

1. Commit all changes in Codespace:

```
git add .
git commit -m "All my work"
git push
```

2. On your local machine:

```
git clone https://github.com/YourUsername/deep-learning-course
cd deep-learning-course
```

3. Set up local environment (see Setup Guide)

4. Continue working locally

11.2 Keeping Codespace as Backup

- Keep your Codespace but stop it
- Work locally most of the time
- Use Codespace only when:
  - Working from different computer
  - Local environment breaks
  - Need more compute power

12 Advanced: Using GPU (Optional)

Note

GitHub Codespaces offers limited free GPU access. This is optional and not required for Week 1.

12.1 Requesting GPU

1. When creating Codespace, click "..." → "Configure dev container"
2. Select machine type with GPU
3. Note: This uses quota faster

12.2 Checking GPU

```
import torch
print(f"CUDA available: {torch.cuda.is_available()}")
if torch.cuda.is_available():
    print(f"GPU: {torch.cuda.get_device_name(0)}")
```

13 Resources

- GitHub Codespaces Docs: `https://docs.github.com/en/codespaces`
- VS Code Tips: `https://code.visualstudio.com/docs/getstarted/tips-and-tricks`
- Git Tutorial: `https://git-scm.com/book/en/v2`
- Course Forum: [link provided by instructor]

14 Getting Help

If you have issues with Codespaces:

1. Check GitHub Codespaces documentation
2. Post in course forum
3. Email instructor with:
    - Screenshot of error
    - What you were trying to do
    - Your repository URL
4. Come to office hours

15 Summary

1. Create GitHub account
2. Fork/create course repository
3. Create Codespace
4. Set up virtual environment
5. Install packages
6. Work in Jupyter notebooks
7. **Commit and push regularly!**
8. Stop Codespace when done
9. Transition to local setup when possible

Important

Remember: Codespaces is a **temporary solution**. Work on getting your local environment set up for the long term!