Introduction to Deep Learning
Local Development Environment Setup Guide
Windows

MSc Artificial Intelligence

Contents

1 Overview

This guide will help you set up a local Python development environment on **Windows** for the Deep Learning course. You will:

- Install Python 3.10 or later
- Create a virtual environment
- Install PyTorch and required packages
- Verify your installation with Jupyter Notebook

Important

Please complete this setup **before** the first exercise session. If you encounter problems, consult the troubleshooting guide or use GitHub Codespaces as a temporary backup.

Note

Estimated time: 30-45 minutes depending on your internet speed.

2 Step 1: Check Existing Python Installation

First, check if you already have Python 3.10 or later installed.

2.1 Open Command Prompt

Press `Win + R`, type `cmd`, and press Enter.

2.2 Check Python Version

In Command Prompt, run:

```
python --version
```

If you see `Python 3.10.x` or higher:

- Great! Skip to Step 3 (Setting Up Virtual Environment)

If you see an error or older version:

- Continue to Step 2 to install Python

3 Step 2: Install Python

3.1 Download Python

1. Go to `https://www.python.org/downloads/`
2. Click on “Download Python 3.11.x” (or the latest 3.x version)
3. The download should start automatically
4. Save the installer file (e.g., `python-3.11.x-amd64.exe`)

3.2 Run the Installer

1. Locate the downloaded installer and double-click to run it
2. **CRITICAL STEP:** At the bottom of the installer window, check the box that says:

   `Add Python 3.11 to PATH`

3. Click “Install Now”
4. Wait for installation to complete (2-5 minutes)
5. Click “Close” when finished

⚠ Important

The “Add Python to PATH” checkbox is crucial! Without this, Python commands won’t work from the command line. If you forgot to check this box, uninstall Python and reinstall with this option checked.

3.3 Verify Installation

1. Open a **new** Command Prompt (close the old one if still open)
2. Run:

   ```
   python --version
   ```

3. You should see:

   ```
   Python 3.11.x
   ```

4. Also verify pip is installed:

   ```
   pip --version
   ```

💡 Tip

If you still get “’python’ is not recognized”, see the Troubleshooting Guide section on PATH issues.

4 Step 3: Create Project Folder

4.1 Navigate to Documents

In Command Prompt:

```
cd C:\Users\%USERNAME%\Documents
```

4.2 Create and Navigate to Project Folder

```
mkdir DeepLearning
cd DeepLearning
```

ℹ Note

You can create the folder anywhere, but avoid:

- `C:\Program Files` (permission issues)
- Folders with spaces or special characters in the name
- Very deep nested paths (Windows path length limits)

5 Step 4: Create Virtual Environment

A virtual environment keeps your project dependencies isolated.

5.1 Create the Virtual Environment

In your `DeepLearning` folder:

```
python -m venv deep_learning_env
```

This creates a new folder called `deep_learning_env` containing:

- Python interpreter
- pip package manager
- Space for installed packages

Wait 30-60 seconds for creation to complete.

5.2 Activate the Virtual Environment

5.2.1 If using Command Prompt (cmd):

```
deep_learning_env\Scripts\activate.bat
```

5.2.2 If using PowerShell:

```
deep_learning_env\Scripts\Activate.ps1
```

⚠ Important

If PowerShell gives an error about execution policy, see the Troubleshooting Guide. Use Command Prompt (cmd) as an alternative.

5.3 Verify Activation

After activation, you should see:

```
(deep_learning_env) C:\Users\YourName\Documents\DeepLearning>
```

The (deep_learning_env) prefix indicates the virtual environment is active!

Tip

To deactivate later, simply type:

```
deactivate
```

6 Step 5: Install Required Packages

Important

Make sure your virtual environment is activated! You should see (deep_learning_env) in your prompt.

6.1 Upgrade pip

First, upgrade pip to the latest version:

```
python -m pip install --upgrade pip
```

6.2 Download requirements.txt

1. Download `requirements.txt` from the course repository
2. Save it to your `DeepLearning` folder
3. Verify it's there:

   ```
   dir requirements.txt
   ```

6.3 Install All Packages

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

Note

This will take 5-10 minutes as PyTorch is a large package (700MB). You'll see progress bars for each package being downloaded and installed.

6.4 Alternative: Manual Installation

If you don't have `requirements.txt`, install packages individually:

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

Tip

If installation is very slow, see the Troubleshooting Guide for tips on using PyTorch's CDN.

7 Step 6: Verify Installation

7.1 Test Python and PyTorch

1. Start Python interpreter:

```
python
```

2. In the Python prompt (>>>), run:

```
import torch
print(f"PyTorch version: {torch.__version__}")
print(f"CUDA available: {torch.cuda.is_available()}")

# Create a test tensor
x = torch.tensor([1, 2, 3])
print(f"Test tensor: {x}")
```

3. You should see:

```
PyTorch version: 2.x.x
CUDA available: False
Test tensor: tensor([1, 2, 3])
```

4. Exit Python:

```
exit()
```

Note

`CUDA available: False` is normal if you don't have an NVIDIA GPU. PyTorch will use your CPU, which is fine for this course.

7.2 Launch Jupyter Notebook

1. Make sure your virtual environment is still activated

2. Run:

```
jupyter notebook
```

3. A browser window should open automatically showing the Jupyter interface

4. You should see your `DeepLearning` folder contents

💡 Tip

If the browser doesn't open automatically:

- Look for a URL in the terminal output
- It will look like: `http://localhost:8888/?token=...`
- Copy this URL and paste it into your browser

7.3 Create and Test a Notebook

1. In Jupyter, click "New" → "Python 3 (ipykernel)"
2. In the first cell, type:

```python
import torch
import matplotlib.pyplot as plt
import numpy as np

print("Setup successful!")
print(f"PyTorch version: {torch.__version__}")

# Simple test
x = torch.randn(5)
print(f"Random tensor: {x}")
```

3. Press `Shift + Enter` to run the cell
4. If you see "Setup successful!" and no errors, everything is working!
5. Close the notebook (File → Close and Halt)
6. Stop Jupyter by pressing `Ctrl + C` twice in the Command Prompt

8 Step 7: Daily Workflow

Every time you want to work on the course:

8.1 Starting Your Work Session

1. Open Command Prompt
2. Navigate to your project folder:

```
cd C:\Users\%USERNAME%\Documents\DeepLearning
```

3. Activate virtual environment:

```
deep_learning_env\Scripts\activate
```

4. Start Jupyter:

```
jupyter notebook
```

5. Work in your notebooks

8.2 Ending Your Work Session

1. Save your notebooks (Ctrl + S)
2. Close notebooks in Jupyter (File → Close and Halt)
3. Stop Jupyter: `Ctrl + C` twice in Command Prompt
4. Deactivate virtual environment:

```
deactivate
```

5. Close Command Prompt

9 Optional: IDE Setup

While Jupyter Notebooks are great for exercises, you may want a full IDE for projects.

9.1 VS Code (Recommended)

9.1.1 Installation

1. Download from `https://code.visualstudio.com/`
2. Run the installer
3. Check "Add to PATH" during installation

9.1.2 Setup for Python

1. Open VS Code
2. Click Extensions icon (left sidebar) or press `Ctrl + Shift + X`
3. Search for and install:
   - "Python" by Microsoft
   - "Jupyter" by Microsoft
4. Open your DeepLearning folder: File → Open Folder
5. Select Python interpreter:
   - Press `Ctrl + Shift + P`
   - Type "Python: Select Interpreter"
   - Choose the one in `deep_learning_env\Scripts\python.exe`
6. You can now open and run .ipynb files directly in VS Code!

9.2 PyCharm Community (Alternative)

1. Download from `https://www.jetbrains.com/pycharm/download/#section=windows`
2. Choose "Community Edition" (free)
3. Run installer
4. Open your DeepLearning folder
5. Configure interpreter:
   - File → Settings → Project → Python Interpreter
   - Click gear icon → Add
   - Choose "Existing environment"
   - Navigate to: `DeepLearning\deep_learning_env\Scripts\python.exe`

10 Optional: GPU Support

> ℹ Note
>
> GPU support is **not required** for this course, but can speed up training significantly from Week 4 onwards. Skip this section if you don't have an NVIDIA GPU.

10.1 Check if You Have NVIDIA GPU

10.1.1 Method 1: Device Manager

1. Press `Win + X` → Device Manager
2. Expand "Display adapters"
3. Look for NVIDIA GeForce/RTX/Quadro

10.1.2 Method 2: Task Manager

1. Press `Ctrl + Shift + Esc`
2. Click "Performance" tab
3. Look for GPU in the left sidebar

If you don't see NVIDIA GPU, you cannot use CUDA. Skip the rest of this section.

10.2 Install NVIDIA Drivers

1. Go to `https://www.nvidia.com/download/index.aspx`
2. Select your GPU model
3. Download and install the latest driver
4. Restart your computer

10.3 Install CUDA Toolkit

1. Go to `https://developer.nvidia.com/cuda-downloads`
2. Select: Windows → x86_64 → 11 → exe (local)
3. Download CUDA Toolkit 11.8 or 12.1
4. Run the installer (will take 10-15 minutes)
5. Follow default options

10.4 Reinstall PyTorch with CUDA

1. Activate your virtual environment
2. Uninstall current PyTorch:

```
pip uninstall torch torchvision torchaudio
```

3. Install PyTorch with CUDA 11.8:

```
pip install torch torchvision torchaudio --index-url https://
    download.pytorch.org/whl/cu118
```

4. Or for CUDA 12.1:

```
pip install torch torchvision torchaudio --index-url https://
    download.pytorch.org/whl/cu121
```

10.5 Verify GPU Support

```python
import torch

print(f"CUDA available: {torch.cuda.is_available()}")

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

    # Test GPU computation
    x = torch.randn(3, 3).cuda()
    print(f"Tensor on GPU: {x.device}")
```

Should print:

```
CUDA available: True
GPU: NVIDIA GeForce RTX 3060
CUDA version: 11.8
Tensor on GPU: cuda:0
```

⚠ Important

GPU setup can be tricky. If you have issues, stick with CPU for now. We can revisit GPU setup in Week 4 when it becomes more beneficial.

11 Troubleshooting Common Issues

11.1 Python Not Recognized

Problem

Error: `'python' is not recognized as an internal or external command`

✓ Solution

Python is not in your PATH:

Option 1: Reinstall Python

1. Uninstall current Python
2. Reinstall and check "Add Python to PATH"

Option 2: Manually add to PATH

1. Search for "Environment Variables" in Start Menu
2. Click "Environment Variables"
3. Under "System variables", find "Path", click "Edit"
4. Click "New" and add:
   - `C:\Users\YourUsername\AppData\Local\Programs\Python\Python311`
   - `C:\Users\YourUsername\AppData\Local\Programs\Python\Python311\Scripts`
5. Click OK, close all terminals, open a new Command Prompt

11.2 Virtual Environment Won't Activate

Problem

After running activation command, no `(deep_learning_env)` prefix appears.

Solution

PowerShell Issue:

1. PowerShell may block scripts by default
2. Run PowerShell as Administrator
3. Execute:

   ```
   Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope
      CurrentUser
   ```

4. Try activating again:

   ```
   deep_learning_env\Scripts\Activate.ps1
   ```

Alternative: Use Command Prompt

```
deep_learning_env\Scripts\activate.bat
```

11.3 Permission Denied

Problem

Getting "Access is denied" or permission errors.

Solution

1. Don't create folder in `C:\Program Files`
2. Use your user directory: `C:\Users\YourUsername\Documents`
3. Run Command Prompt as regular user (not Administrator)
4. Check antivirus isn't blocking Python

11.4 Package Installation Fails

Problem

`pip install` fails with timeout or connection errors.

✓ Solution

Try PyTorch CDN (faster):

```
pip install torch torchvision torchaudio --index-url https://
    download.pytorch.org/whl/cpu
```

Increase timeout:

```
pip install --default-timeout=1000 torch
```

Check firewall/antivirus:

- Temporarily disable antivirus
- Add Python and pip to firewall exceptions

11.5 Jupyter Notebook Won't Start

🐛 Problem

`jupyter notebook` fails or browser doesn't open.

✓ Solution

1. Verify Jupyter is installed:

```
pip install jupyter notebook
```

2. Try different port:

```
jupyter notebook --port=8889
```

3. Check if port 8888 is in use:

```
netstat -ano | findstr :8888
```

4. Manually copy URL from terminal to browser

11.6 Module Not Found in Jupyter

🐛 Problem

`ModuleNotFoundError: No module named 'torch'` in Jupyter, even though you installed it.

✓ Solution

Jupyter is using wrong Python environment!

1. Make sure virtual environment was activated BEFORE starting Jupyter
2. Check Python path in notebook:

   ```
   import sys
   print(sys.executable)
   ```

   Should point to `deep_learning_env`
3. Register environment as kernel:

   ```
   python -m ipykernel install --user --name=deep_learning_env
   ```

   Then: Kernel → Change kernel → deep_learning_env

11.7 Long Path Errors

🐞 Problem

Errors about file paths being too long.

✓ Solution

Enable long paths in Windows:

1. Open Registry Editor (Win+R, type `regedit`)
2. Navigate to: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem`
3. Set `LongPathsEnabled` to 1
4. Restart computer

Or use shorter path:

```
cd C:\DL
```

11.8 Antivirus Blocking Installation

🐞 Problem

Installation hangs or fails with cryptic errors.

✓ Solution

1. Temporarily disable antivirus during installation
2. Add Python directory to antivirus exceptions
3. Add pip to exceptions
4. Consider using Windows Defender instead of third-party antivirus

12 Getting Help

If you've tried the troubleshooting steps and still have issues:

1. Document your error:
   - Copy full error message
   - Note your Windows version
   - Note Python version
   - What you were trying to do
   - What you've already tried
2. Get help:
   - Post in course forum with documentation
   - Email instructor with details
   - Come to office hours
3. Temporary workaround:
   - Use GitHub Codespaces (see separate guide)
   - Continue with exercises while troubleshooting local setup

13 Next Steps

1. Download Week 1 exercise notebooks from course repository
2. Place them in your `DeepLearning` folder
3. Activate virtual environment
4. Start Jupyter Notebook
5. Open `week1_exercises_starter.ipynb`
6. You're ready to start coding!

💡 Tip

Create a habit: Always activate your virtual environment before working on course materials!

14 Quick Reference

14.1 Essential Commands

```
# Navigate to project
cd C:\Users\%USERNAME%\Documents\DeepLearning

# Activate virtual environment
deep_learning_env\Scripts\activate

# Start Jupyter
jupyter notebook
```

```
# Stop Jupyter
# Press Ctrl+C twice

# Deactivate virtual environment
deactivate
```

14.2 Troubleshooting Quick Fixes

- **Python not recognized**: Reinstall with “Add to PATH” checked
- **Virtual env won’t activate**: Use `activate.bat` instead of `Activate.ps1`
- **Module not found in Jupyter**: Make sure venv was activated before starting Jupyter
- **Slow installation**: Use PyTorch CDN (see Troubleshooting Guide)