

Introduction to Deep Learning

Local Development Environment Setup Guide

MSc Computer Science

Week 1

Contents

1 Overview	3
2 Prerequisites	3
2.1 Check Python Installation	3
2.1.1 Windows	3
2.1.2 Linux	3
2.1.3 macOS	3
3 Installing Python	3
3.1 Windows	3
3.2 Linux (Ubuntu/Debian)	4
3.3 macOS	4
3.3.1 Option 1: Using Homebrew (Recommended)	4
3.3.2 Option 2: Official Installer	4
4 Setting Up Virtual Environment	4
4.1 Windows	5
4.2 Linux	5
4.3 macOS	5
5 Installing Required Packages	6
5.1 Download requirements.txt	6
5.2 Install Packages	6
5.2.1 Windows	6
5.2.2 Linux/macOS	6
5.3 Manual Installation (if requirements.txt not available)	6
6 Verifying Installation	6
6.1 Check PyTorch Installation	6
6.2 Launch Jupyter Notebook	7
6.3 Test Notebook	7
7 Daily Workflow	7
7.1 Windows	8
7.2 Linux/macOS	8
8 IDE Setup (Optional but Recommended)	8
8.1 VS Code (Recommended)	8
8.2 PyCharm (Alternative)	8

9 GPU Support (Optional)	9
9.1 Check if you have NVIDIA GPU	9
9.1.1 Windows	9
9.1.2 Linux	9
9.1.3 macOS	9
9.2 Installing CUDA Support (NVIDIA GPU only)	9
9.3 Apple Silicon (M1/M2/M3) Support	10
10 Getting Help	10
11 Next Steps	10

1 Overview

This guide will help you set up a local Python development environment 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.

ℹ Note

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

2 Prerequisites

2.1 Check Python Installation

First, check if you have Python 3.10 or later installed:

2.1.1 Windows

Open Command Prompt (Win + R, type `cmd`, press Enter) and run:

```
python --version
```

2.1.2 Linux

Open Terminal (Ctrl + Alt + T) and run:

```
python3 --version
```

2.1.3 macOS

Open Terminal (Cmd + Space, type “Terminal”, press Enter) and run:

```
python3 --version
```

If you see Python 3.10.x or higher, you’re good to go! Otherwise, proceed to Section 3.

3 Installing Python

3.1 Windows

1. Go to <https://www.python.org/downloads/>
2. Download Python 3.11 or later (the latest stable version)

3. Run the installer
4. **IMPORTANT:** Check “Add Python to PATH” before clicking Install
5. Click “Install Now”
6. Verify installation by opening a new Command Prompt and running:

```
python --version
```

Important

Make sure to check “Add Python to PATH”! Without this, Python commands won’t work from the command line.

3.2 Linux (Ubuntu/Debian)

Python 3 is usually pre-installed. If not, or if you need a newer version:

```
# Update package list
sudo apt update

# Install Python 3.11
sudo apt install python3.11 python3.11-venv python3-pip

# Verify installation
python3.11 --version
```

For other distributions, use your package manager (dnf, pacman, etc.).

3.3 macOS

3.3.1 Option 1: Using Homebrew (Recommended)

If you have Homebrew installed:

```
# Install Python
brew install python@3.11

# Verify installation
python3 --version
```

3.3.2 Option 2: Official Installer

1. Go to <https://www.python.org/downloads/macos/>
2. Download the macOS installer for Python 3.11 or later
3. Run the installer package
4. Verify installation in Terminal:

```
python3 --version
```

4 Setting Up Virtual Environment

Virtual environments keep your project dependencies isolated and manageable.

4.1 Windows

1. Open Command Prompt
2. Navigate to where you want to create your project folder:

```
cd C:\Users\YourUsername\Documents
mkdir DeepLearning
cd DeepLearning
```

3. Create virtual environment:

```
python -m venv deep_learning_env
```

4. Activate the virtual environment:

```
deep_learning_env\Scripts\activate
```

5. You should see (deep_learning_env) at the beginning of your command prompt

💡 Tip

To deactivate the virtual environment later, simply type `deactivate`.

4.2 Linux

1. Open Terminal
2. Navigate to your preferred location:

```
cd ~/Documents
mkdir DeepLearning
cd DeepLearning
```

3. Create virtual environment:

```
python3 -m venv deep_learning_env
```

4. Activate the virtual environment:

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

5. You should see (deep_learning_env) at the beginning of your prompt

4.3 macOS

1. Open Terminal
2. Navigate to your preferred location:

```
cd ~/Documents
mkdir DeepLearning
cd DeepLearning
```

3. Create virtual environment:

```
python3 -m venv deep_learning_env
```

4. Activate the virtual environment:

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

5. You should see (deep_learning_env) at the beginning of your prompt

5 Installing Required Packages

Important

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

5.1 Download requirements.txt

Download the requirements.txt file from the course repository or create it yourself (see Section ??).

5.2 Install Packages

With your virtual environment activated, run:

5.2.1 Windows

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

5.2.2 Linux/macOS

```
python3 -m pip install --upgrade pip
pip install -r requirements.txt
```

Note

Installation may take 5-10 minutes depending on your internet connection. PyTorch is a large package (700MB).

5.3 Manual Installation (if requirements.txt not available)

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

6 Verifying Installation

6.1 Check PyTorch Installation

Run Python in your terminal:

```
# Windows:
python

# Linux/macOS:
python3
```

Then in the Python interpreter:

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

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

You should see the PyTorch version and a tensor printed. Type `exit()` to leave Python.

Note

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

6.2 Launch Jupyter Notebook

With your virtual environment activated:

```
jupyter notebook
```

This should open a web browser with the Jupyter interface. You should see your **DeepLearning** folder.

Tip

If the browser doesn't open automatically, look for a URL in the terminal output (usually `http://localhost:8888/...`) and copy it into your browser.

6.3 Test Notebook

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

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

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

3. Press Shift + Enter to run the cell
4. If you see “Setup successful!” and the PyTorch version, everything is working!

7 Daily Workflow

Every time you want to work on the course:

7.1 Windows

```
# Navigate to your project folder
cd C:\Users\YourUsername\Documents\DeepLearning

# Activate virtual environment
deep_learning_env\Scripts\activate

# Start Jupyter
jupyter notebook

# When done, deactivate
deactivate
```

7.2 Linux/macOS

```
# Navigate to your project folder
cd ~/Documents/DeepLearning

# Activate virtual environment
source deep_learning_env/bin/activate

# Start Jupyter
jupyter notebook

# When done, deactivate
deactivate
```

8 IDE Setup (Optional but Recommended)

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

8.1 VS Code (Recommended)

1. Download from <https://code.visualstudio.com/>
2. Install the Python extension (Microsoft)
3. Install the Jupyter extension (Microsoft)
4. Open your project folder
5. Select your virtual environment:
 - Press Ctrl/Cmd + Shift + P
 - Type “Python: Select Interpreter”
 - Choose the one in `deep_learning_env`

8.2 PyCharm (Alternative)

1. Download PyCharm Community Edition from <https://www.jetbrains.com/pycharm/>
2. Open your project folder
3. Configure interpreter: File → Settings → Project → Python Interpreter
4. Select your `deep_learning_env`

9 GPU Support (Optional)

Note

GPU support is **not required** for this course, but can speed up training significantly from Week 4 onwards.

9.1 Check if you have NVIDIA GPU

9.1.1 Windows

1. Right-click on desktop → Display settings → Advanced display
2. Or: Open Task Manager → Performance tab

9.1.2 Linux

```
lspci | grep -i nvidia
```

9.1.3 macOS

Recent Macs use Apple Silicon (M1/M2/M3), which has different support. For Intel Macs, check: About This Mac → Graphics.

9.2 Installing CUDA Support (NVIDIA GPU only)

1. Install NVIDIA GPU drivers from <https://www.nvidia.com/drivers>
2. Install CUDA Toolkit 11.8 or 12.1 from <https://developer.nvidia.com/cuda-downloads>
3. Reinstall PyTorch with CUDA support:

```
# Uninstall current PyTorch
pip uninstall torch torchvision torchaudio

# Install with CUDA 11.8 (check PyTorch website for latest)
pip install torch torchvision torchaudio --index-url https://
download.pytorch.org/whl/cu118
```

4. Verify:

```
import torch
print(torch.cuda.is_available()) # Should print True
print(torch.cuda.get_device_name(0)) # Your GPU name
```

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.

9.3 Apple Silicon (M1/M2/M3) Support

PyTorch supports Apple's Metal Performance Shaders (MPS):

```
import torch

# Check MPS availability
print(f"MPS available: {torch.backends.mps.is_available()}")

# Use MPS device
device = torch.device("mps" if torch.backends.mps.is_available() else "cpu")
x = torch.tensor([1, 2, 3]).to(device)
```

10 Getting Help

If you encounter issues:

1. Check the Troubleshooting Guide (separate document)
2. Ask in the course forum/chat
3. Come to office hours
4. Use GitHub Codespaces as temporary backup (see backup guide)

11 Next Steps

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

Tip

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