Conda & Jupyter Command Reference for Deep Learning

Environment Management

Creating & Managing Environments

bash

Copy

```bash
# Create a new environment with Python
conda create -n myenv python=3.12

# Activate an environment
conda activate myenv

# Deactivate current environment
conda deactivate

# List all environments
conda env list

# Remove an environment
conda env remove -n myenv
```

Environment Files

bash

Copy

```bash
# Export current environment to a file
conda env export > environment.yml

# Export a cleaner, more portable version
conda env export --from-history > environment.yml

# Create environment from file
conda env create -f environment.yml

# Update existing environment from file
conda env update -f environment.yml

# Clone environment with a new name
conda env create -f environment.yml -n new-env-name
```

Package Management

Installing & Updating Packages

bash

Copy

```bash
# Install packages
conda install package_name
conda install package1 package2

# Install from specific channel
conda install -c conda-forge tensorflow

# Install specific version
conda install numpy=1.24

# Update a package
conda update package_name

# Update all packages
conda update --all

# List installed packages
conda list

# List outdated packages
conda list --outdated

# Search for available packages
conda search package_name
```

Jupyter Commands

Starting & Managing Jupyter

bash

Copy

```bash
# Start Jupyter Notebook
jupyter notebook

# Start in a specific directory
jupyter notebook --notebook-dir=/path/to/directory

# Start with a specific port
jupyter notebook --port=9999

# List running notebook servers
jupyter notebook list

# Stop a notebook server (or use Ctrl+C in terminal)
jupyter notebook stop [PORT]
```

Deep Learning Specific

CUDA & GPU Commands

bash

Copy

```bash
# Install TensorFlow with GPU support
conda install -c conda-forge tensorflow

# Install PyTorch with specific CUDA version
conda install -c pytorch pytorch cudatoolkit=11.8

# Check if GPU is available (in Python)
import tensorflow as tf
print("TensorFlow sees GPU:", tf.config.list_physical_devices('GPU'))

import torch
print("PyTorch sees GPU:", torch.cuda.is_available())
```

Common Deep Learning Environment

bash

Copy

```
# Create a standard deep learning environment
cat > environment.yml << 'EOL'
name: deep-learning
channels:
  - conda-forge
  - pytorch
  - defaults
dependencies:
  - python=3.12
  - numpy
  - pandas
  - matplotlib
  - scikit-learn
  - jupyter
  - notebook
  - tensorflow=2.15
  - pytorch=2.1
  - torchvision
  - cudatoolkit=11.8
  - pip
  - pip:
    - transformers
    - opencv-python
EOL

conda env create -f environment.yml
```

Troubleshooting

Common Issues

bash

Copy

```bash
# Clean conda cache (can resolve installation issues)
conda clean --all

# Reinstall a package
conda install --force-reinstall package_name

# Get conda information for support
conda info

# Update conda itself
conda update conda
```

Quick Workflow Reference

bash

Copy

```bash
# Daily startup
conda activate jupyter-env
jupyter notebook

# Installing a new package
conda install new_package
# Update environment file manually, then
conda env export --from-history > environment.yml

# Backing up environment
cp environment.yml environment_$(date +%Y%m%d).yml
```