Machine Learning
Lecture 2

Yang Yuan

Review of the last lecture

- History of AI
  - AI Boom + AI winter, multiple times
  - People started to study many difficult problems long time ago
    - (still unsolved yet…)

Review of the last lecture

Supervised learning: step by step

1. Identify the task you want to solve (e.g., MNIST)
2. Create a dataset, containing thousands, or millions of examples
   - Input $X = (x_1, x_2, \cdots, x_N)$
   - Output $Y = (y_1, y_2, \ldots, y_N)$
3. Define a loss function L to evaluate $f(X)$
4. Learn a function f to minimize L (optimization)
5. **Minimizing L is not enough!** (generalization)
   - We need to ensure f that generalizes well
   - We could use a validation set <u>in practice</u>
   - But <u>in theory</u>, we will see some guarantees

Today's plan

- Create dataset
- Overfit vs underfit
- Unsupervised & semi-supervised learning framework
- Instruction on using pycharm + pytorch

How to create a dataset?

- **Attention**: high quality big data is more important than everything you will learn in this class
  - Everyone can learn how to train a network in 1 hour
    - Easy steps, 10 lines of code
  - Not everyone can build a good dataset!
  - Unfortunately we will not cover it
- How do you build a dataset of millions of data points?

How to create a dataset?

- Create the input X (from internet or other source)
- How to get the correct label Y?
  - Crowd sourcing is the right way to go!
  - https://www.mturk.com/
  - Split the task into millions of micro-tasks
- This is **highly nontrivial**, but super important in practice
  - There will be hundreds or thousands of workers
    - Get paid by their performance
  - How do you know the labels are correct?
    - Usually for any x_i, multiple workers will label it, then take majority voting
    - But sometimes the label is not a simple number!

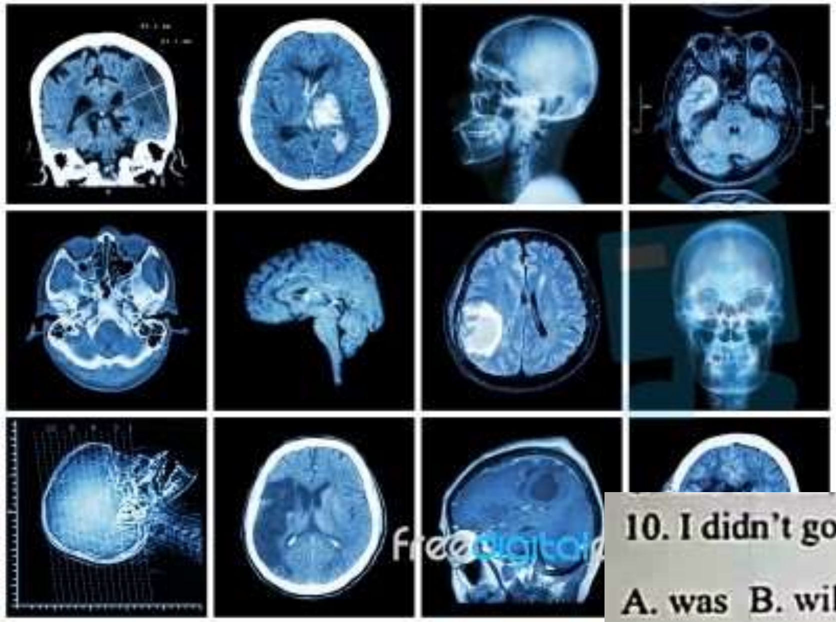
MS COCO dataset

- You need to draw the whole region, which is even harder
- Moreover, how to minimize the cost of labeling?
  - Split the task into multiple small, easy steps
  - Every worker only works on one step (**assembly line**)

ReCAPTCHA: a smart idea

What if labeling tasks are harder?

- For example:
  - Health care
  - Law suits
  - Education
  - ….
- What if some workers are strategic?
  - Spend least time to make most money
  - Use AI to generate label
- Main bottlenect in practice



STATE OF INDIANA) SS: [redacted] CIRCUIT COURT
COUNTY OF [redacted])

LNVN FUNDING, LLC,
Plaintiff,
- vs -
[redacted]
Defendant.

) [redacted] CAUSE NO. [redacted]

ORDER

Comes now the Defendant, [redacted], in person and pro se, and comes now the Plaintiff, LNVN Funding, LLC, either by its representative or by counsel, all for hearing on the Defendant's Motion to Reconsider Pursuant to Indiana Trial Rule 53.4(A).

The Defendant presents argument to the Court.

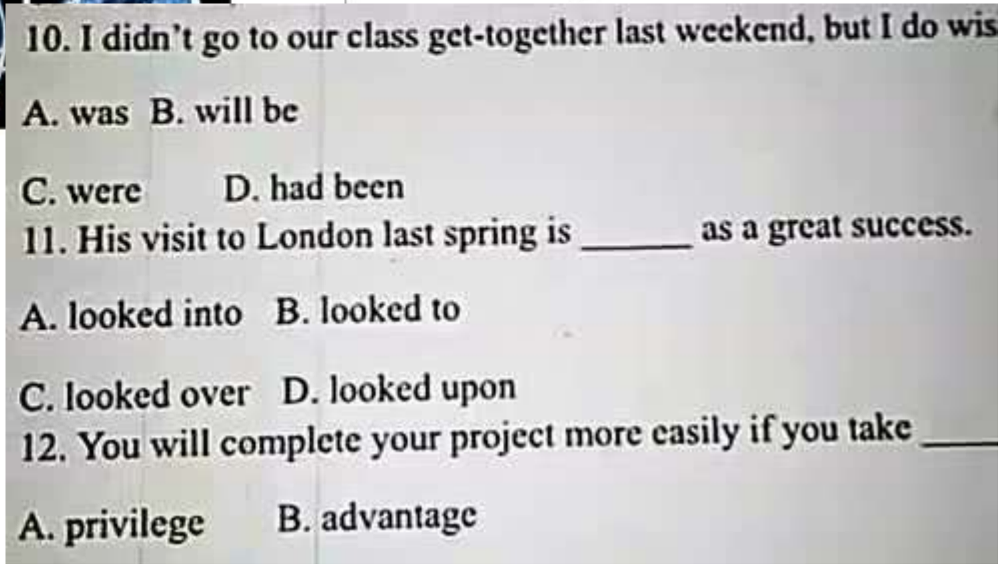
The Court, having heard the argument of the Defendant and being duly advised in the premises, now finds that the Defendant's Motion to Reconsider should be granted and this cause should be dismissed pursuant to Trial Rule 12(B)(6).

IT IS THEREFORE ORDERED AND DECREED by the Court that the Defendant's Motion to Reconsider is hereby granted and this cause is dismissed pursuant to Trial Rule 12(B)(6).

ORDERED THIS 11TH DAY OF MARCH, 2009.

John D. Potter, Judge
[redacted] Court

[redacted]



10. I didn't go to our class get-together last weekend, but I do wis

A. was B. will be

C. were D. had been

11. His visit to London last spring is ______ as a great success.

A. looked into B. looked to

C. looked over D. looked upon

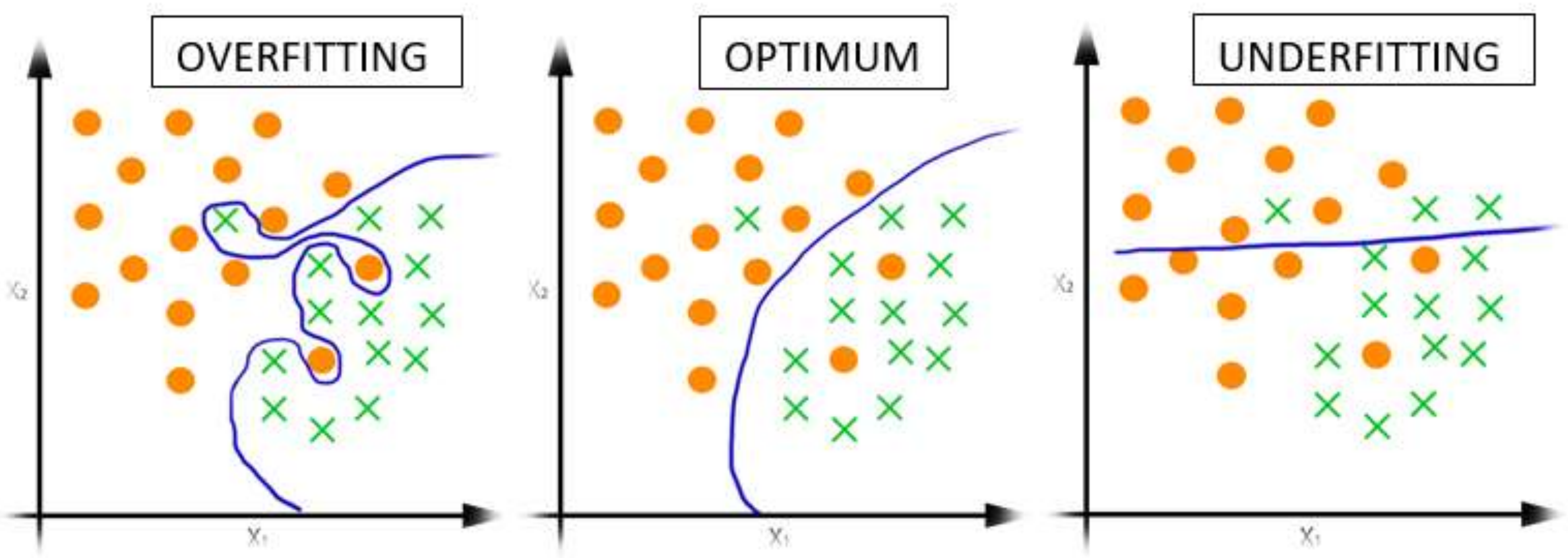
12. You will complete your project more easily if you take ____

A. privilege B. advantage

From now on, always assume we have data

- **Training loss**: $L_{\text{train}} = L(f, X_{train}, Y_{train})$
  - X_{train}, Y_{train} is training set
  - They define the **empirical distribution** $\Pr\big((x_i, y_i)\big) = \frac{1}{N}$
- **Test loss**: $L_{\text{test}} = L(f, X_{test}, Y_{test})$
- **Validation loss** : $L_{valid} = L(f, X_{valid}, Y_{valid})$
  - Practical trick for estimating test loss
- $(X_{test}, Y_{test}), (X_{train}, Y_{train}), (X_{valid}, Y_{valid})$ are all sampled from **population distribution** D_X, D_Y
- **Population loss**: $L_{population} = E_{X,Y \sim D_X, D_Y} L(f, X, Y)$
- We want to minimize $L_{population}$
  - In practice, we estimate $L_{population}$ using L_{test}
  - The ultimate goal in supervised learning

Classical view: overfit vs underfit



Classical view

- Underfit:
  - Your function does not have enough representation power
    - E.g., use a straight line to fit this complicated world
  - Underfit gives you bad L_{train}
  - Usually no generalization problem, because L_{test} is equally bad
- Overfit:
  - Your function had too much representation power
    - E.g., neural network, which can represent everything (we will see later)
  - Overfit easily gives you $L_{train} = 0$
  - However, generalization problem: L_{test} could be very bad!

Classical view

- Since overfitting **could** lead to bad L_{test}
  - Classical view thinks we should avoid it
  - Therefore, we should restrict the representation power of f, so that it could not overfit!
    - This is called "**regularization**"
- Modern view:
  - Sometimes, explicit regularization is not necessary
  - For neural networks, there are implicit regularizations to prevent overfitting
    - Because of optimization process: SGD algorithm
    - We will cover it later
  - In other words, although overfitting could lead to bad L_{test}, it **almost never happens**
  - Researchers were afraid of overfitting for neural networks for decades!

Regularization

- In order to avoid overfitting
  - We want to make sure f is “simple”
    - We will see a few examples later
  - Therefore, if there are two functions:
    - Simple function f with worse L_{train}
    - Complex function f with better L_{train}
  - Maybe we should pick the first one!
- A more fundamental fact:
  - The function that minimizes empirical loss (i.e., L_{train})
  - Is not necessarily the one that minimizes population loss (i.e., L_{test})
  - This is true even in convex case [stochastic convex optimization, SSSS09]

投票 最多可选1项

设置

Is it clear? Do we need to go back?

A It is clear

B Let us go back

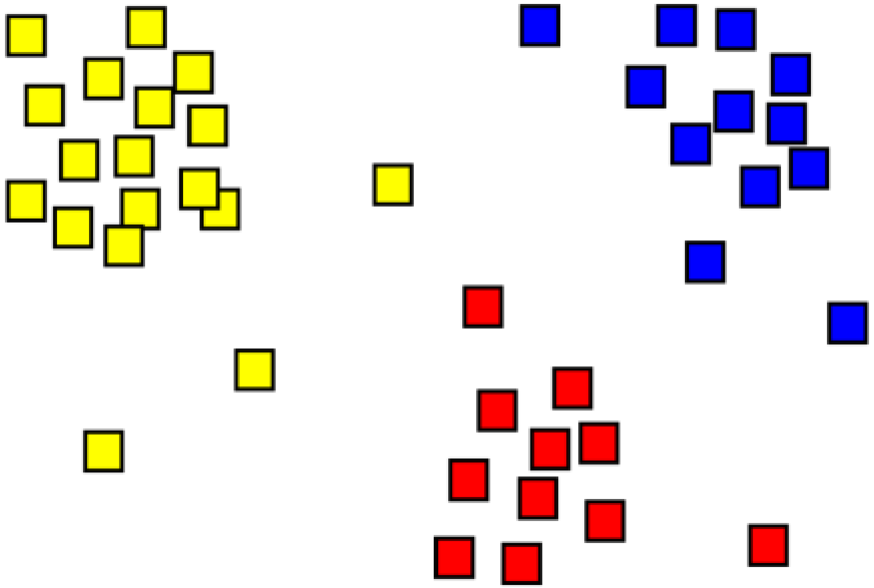
提交

Unsupervised learning framework

- Supervised learning framework
  - Inputs X
  - Labels Y
- Unsupervised learning framework
  - Inputs X
  - No labels Y
- What can you do?
  - Learn the distribution of X

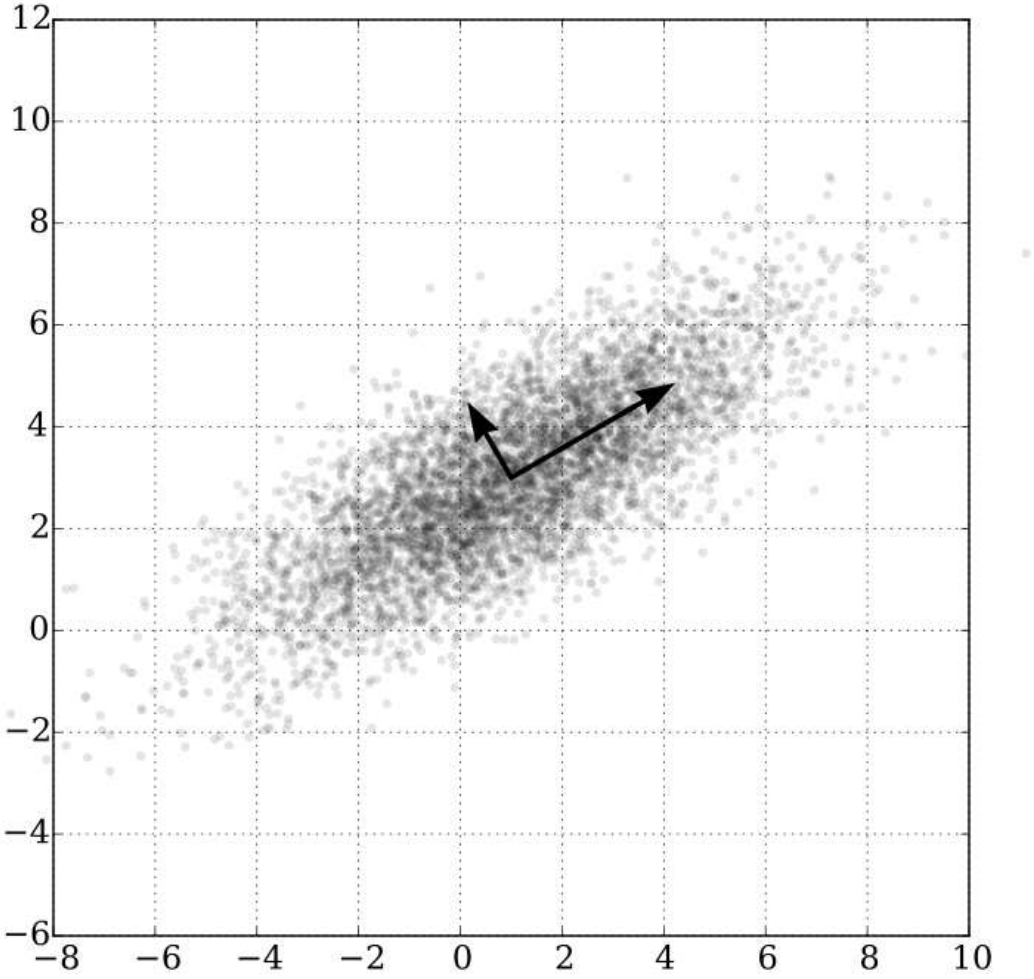
1. Clustering

- Objects in the same group are similar to each other
- But no unique solution!
  - Depends on loss function
- Why is it useful?
  - Data mining
    - To see whether inside the same cluster they share something in common
  - Speed-up optimization process
  - Recommendation system



2. Principle component analysis

- Find most important components (directions)
- a best-fitting line is defined as one that minimizes the average squared distance from the points to the line.
- Or, the line with most “variance”

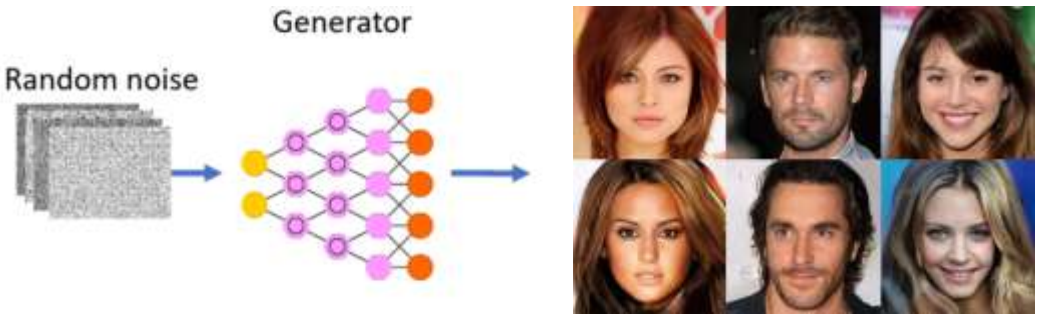


3. Generative model

- Generative model, is essentially what God has, what we really want
- Why do we want to describe the data distribution?
  - Once we have the (true) data distribution, we can sample it and get abundant data!

- Data distribution is hard to describe
  - Usually: X follows Gaussian?
  - Empirically: X follows a bizarre distribution

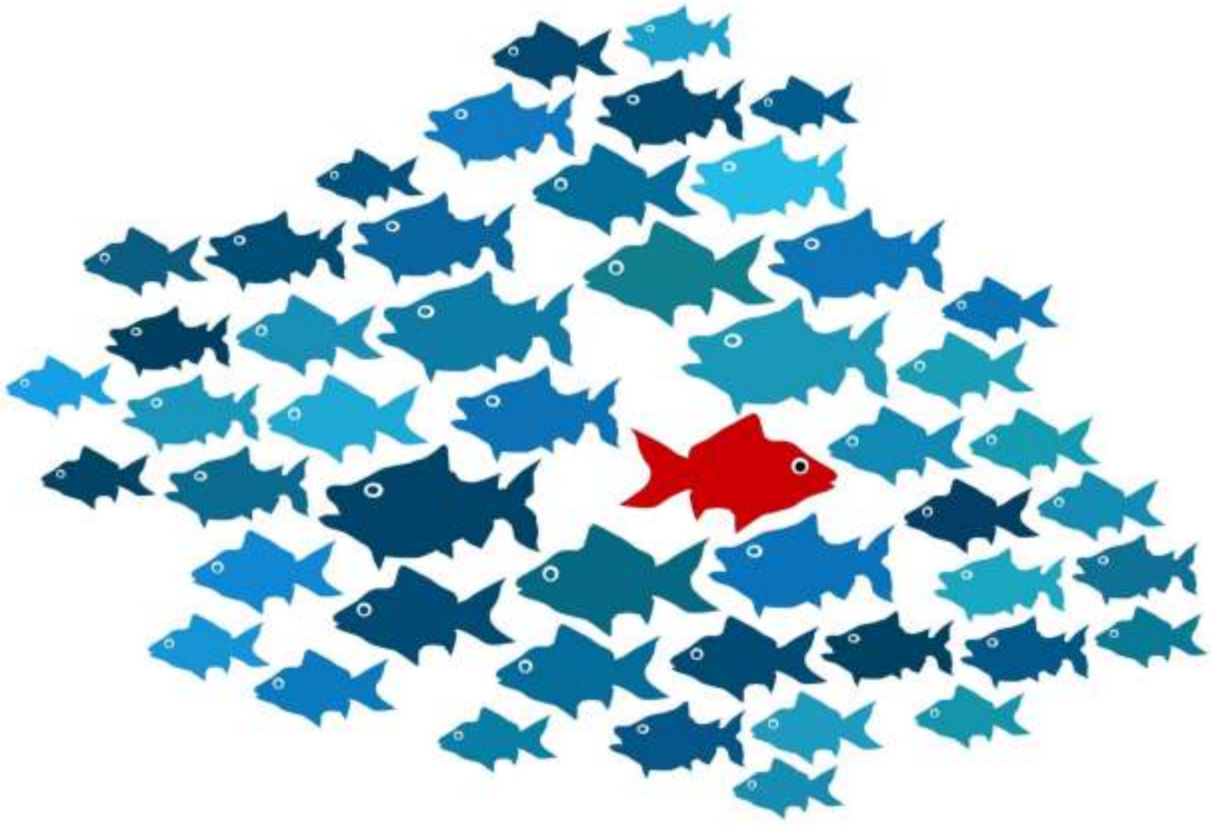
Why generative model?



- How do we describe a bizarre distribution?
  - Generative model!
- Generative model did not directly answer the question of “describing a bizarre distribution”
- Instead, it maps a Gaussian to the target distribution
- Hope:
  - When sampling the Gaussian, after the transformation, we get a good sample from real data X
- So we get “real” faces for free!
- Bonus: we also get to understand the hidden structure of X
  - What does it mean?

4. Anomaly detection

- Find rare items/events, that are different from other data points



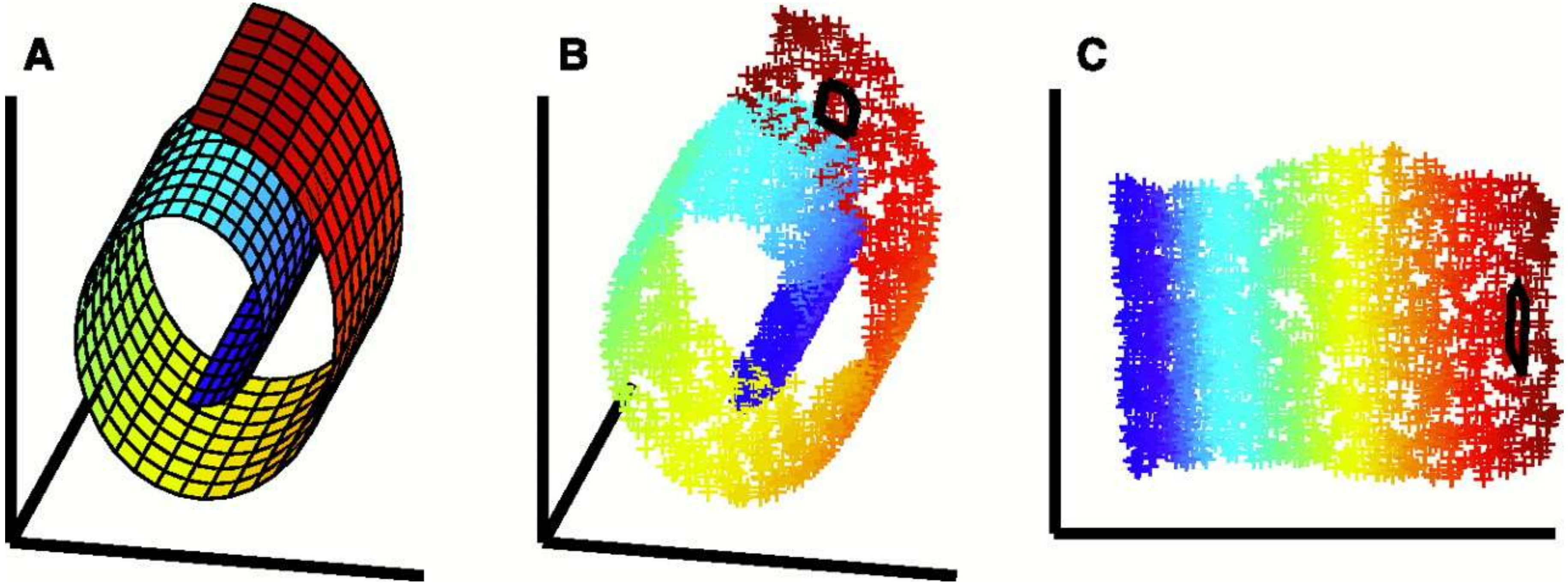
4. Anomaly detection

- Very important
  - COVID-19 auto detection
  - Weird online transactions
  - Computer virus monitoring

- However, how to detect anomalies?
  - Hard to give clear definitions…
  - Diverse solutions

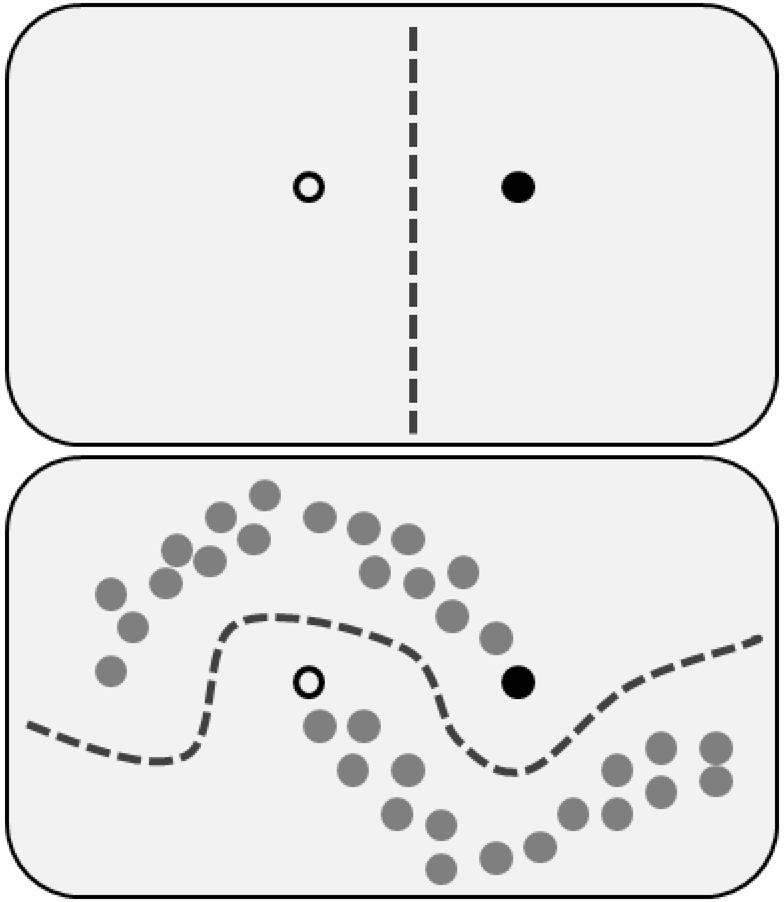
5. Dimension reduction

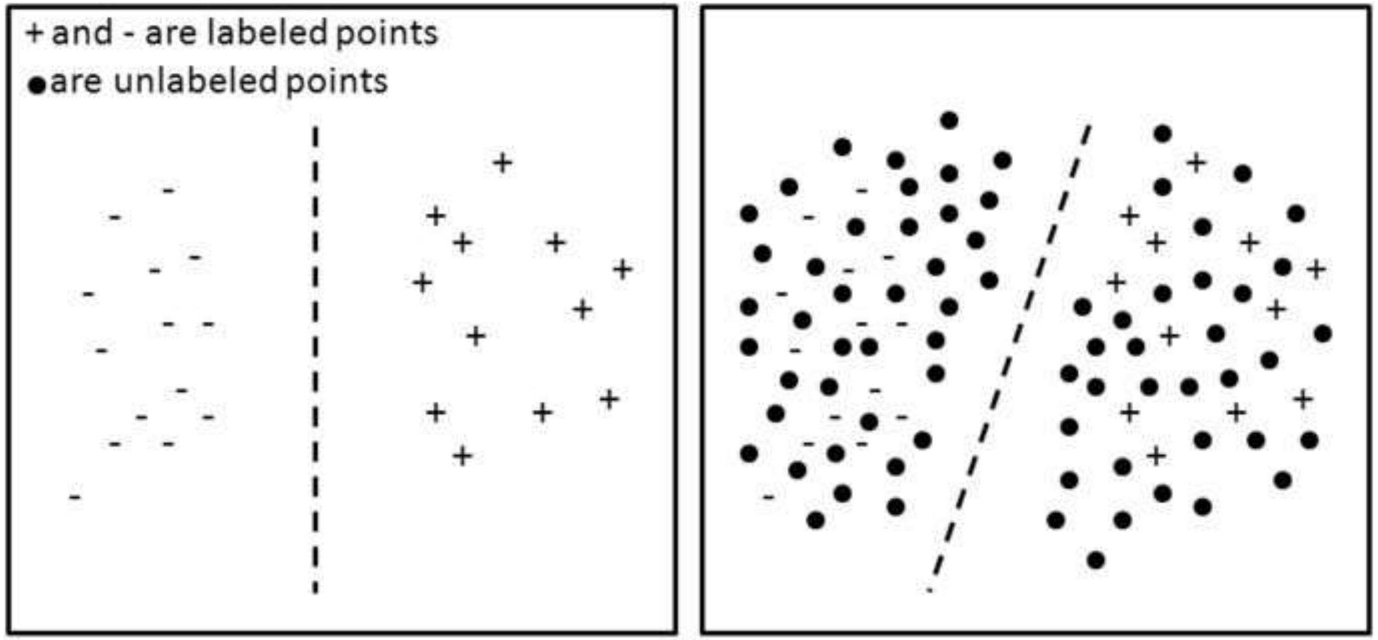
- Some data points live in high dim space, but are inherently low dim.
- How to map them into low dim? (E.g., PCA is one kind of dimension reduction)



Semi-supervised learning

- In practice, semi-supervised learning is common
  - Some data points (say 10%) have labels
  - Some data points (say 90%) do not have labels
- Can we always use the unlabeled data to improve prediction?
  - Not always! Why? (Y is pure noise)
- (Implicitly), we need some assumptions.
  - Continuity assumption
    - Points which are close to each other are more likely to share a label
    - Gives geometrically simple decision boundaries
  - Manifold assumption
    - The data lie approximately on a manifold of much lower dimension than the input space





(a)

(b)

投票 最多可选1项

设置

Is it clear? Do we need to go back?

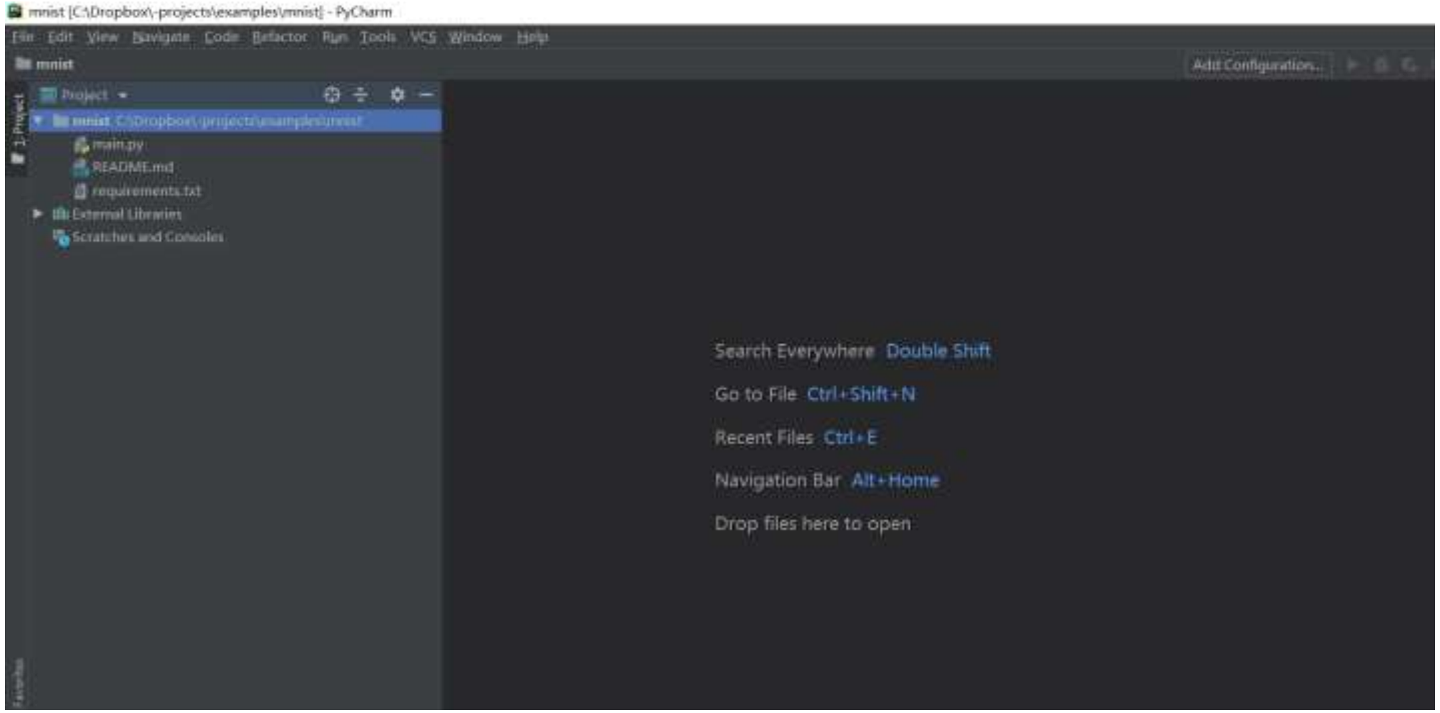
A It is clear

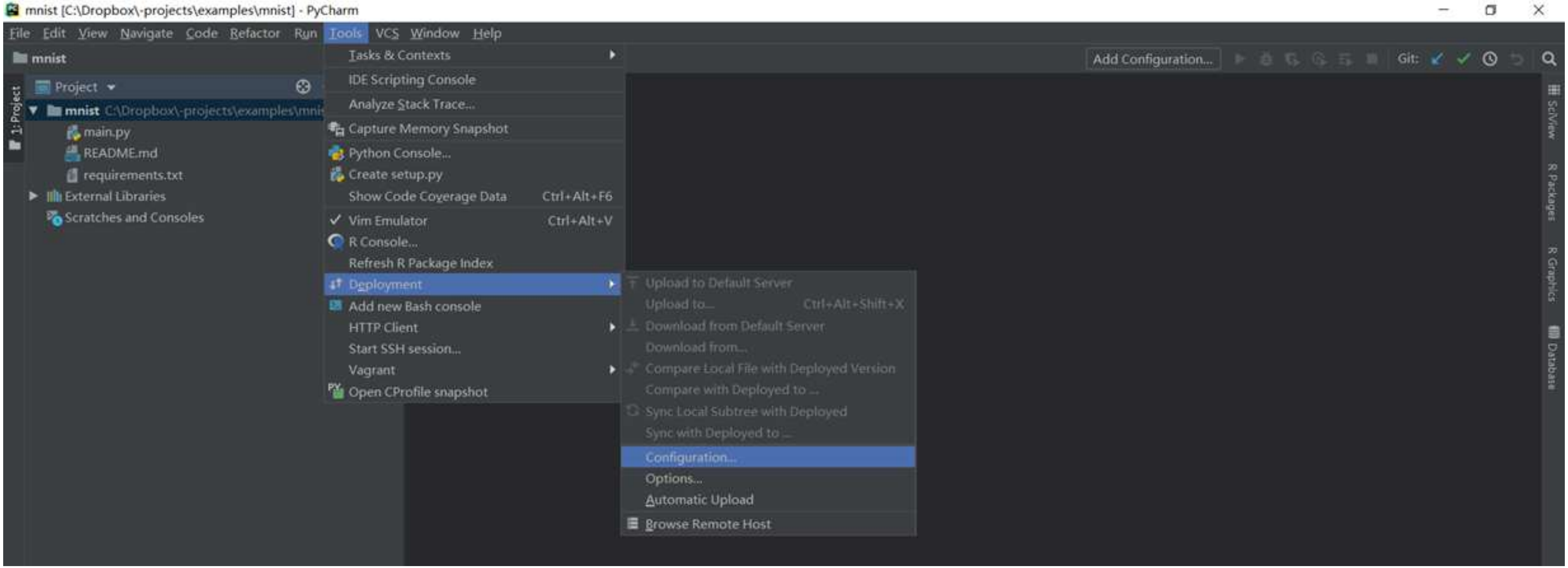
B Let us go back

提交

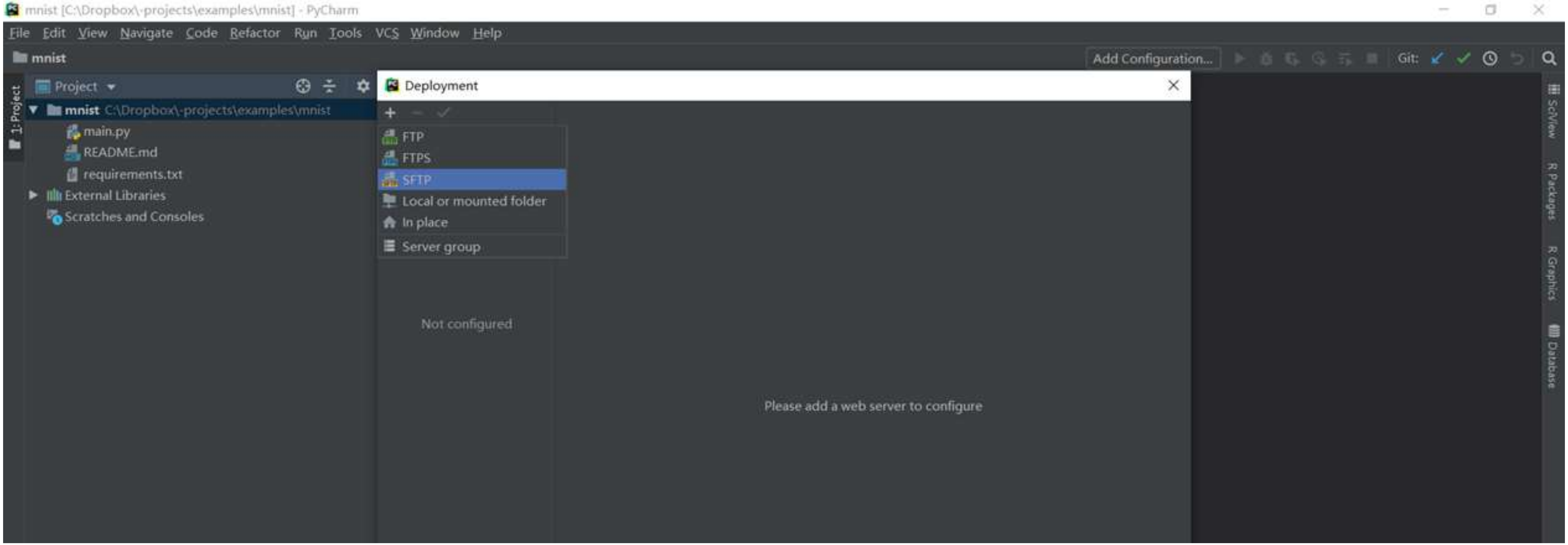
Use github

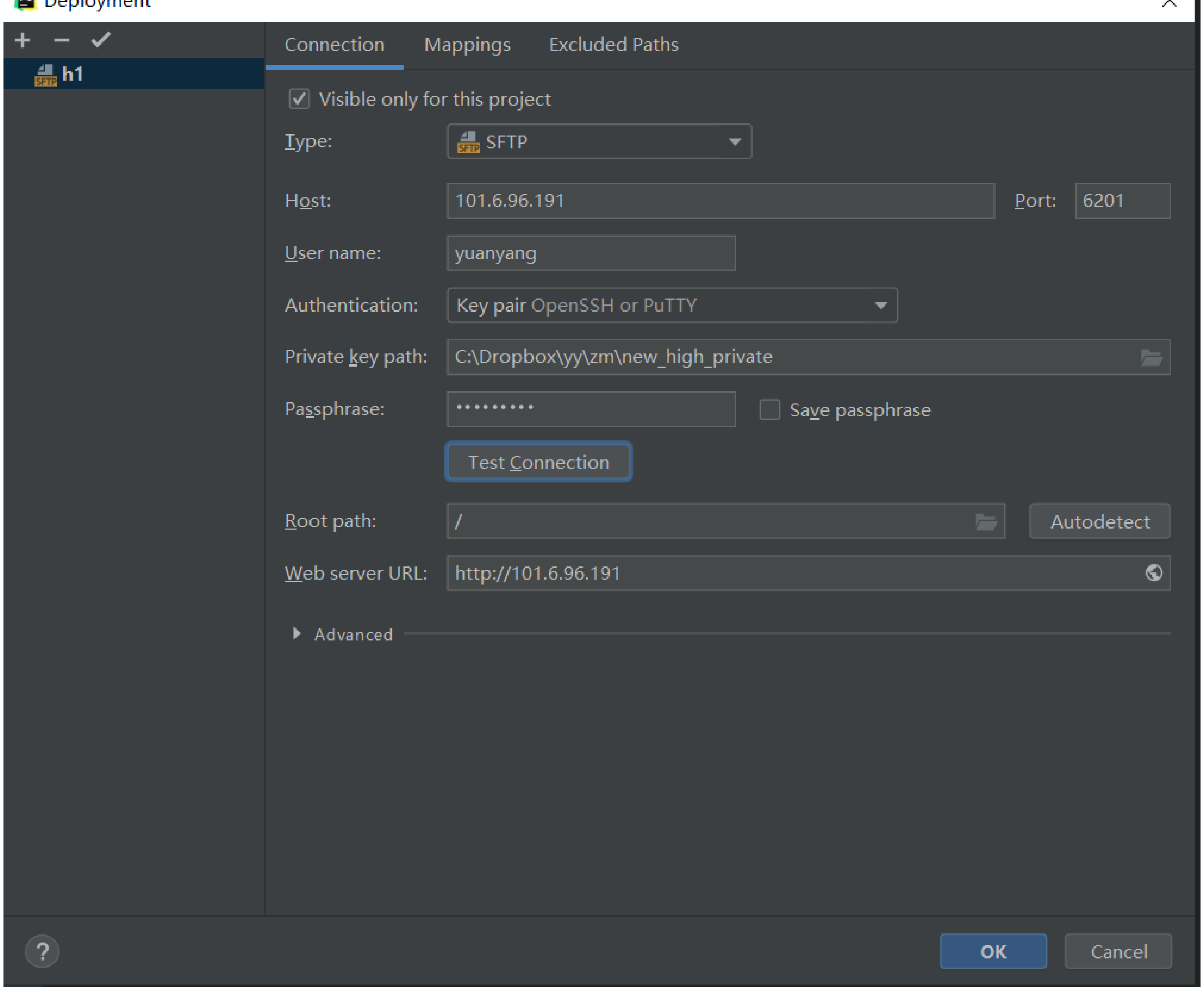
- git clone https://github.com/pytorch/examples.git
  - Download the source code for pytorch examples
- Open project using pycharm



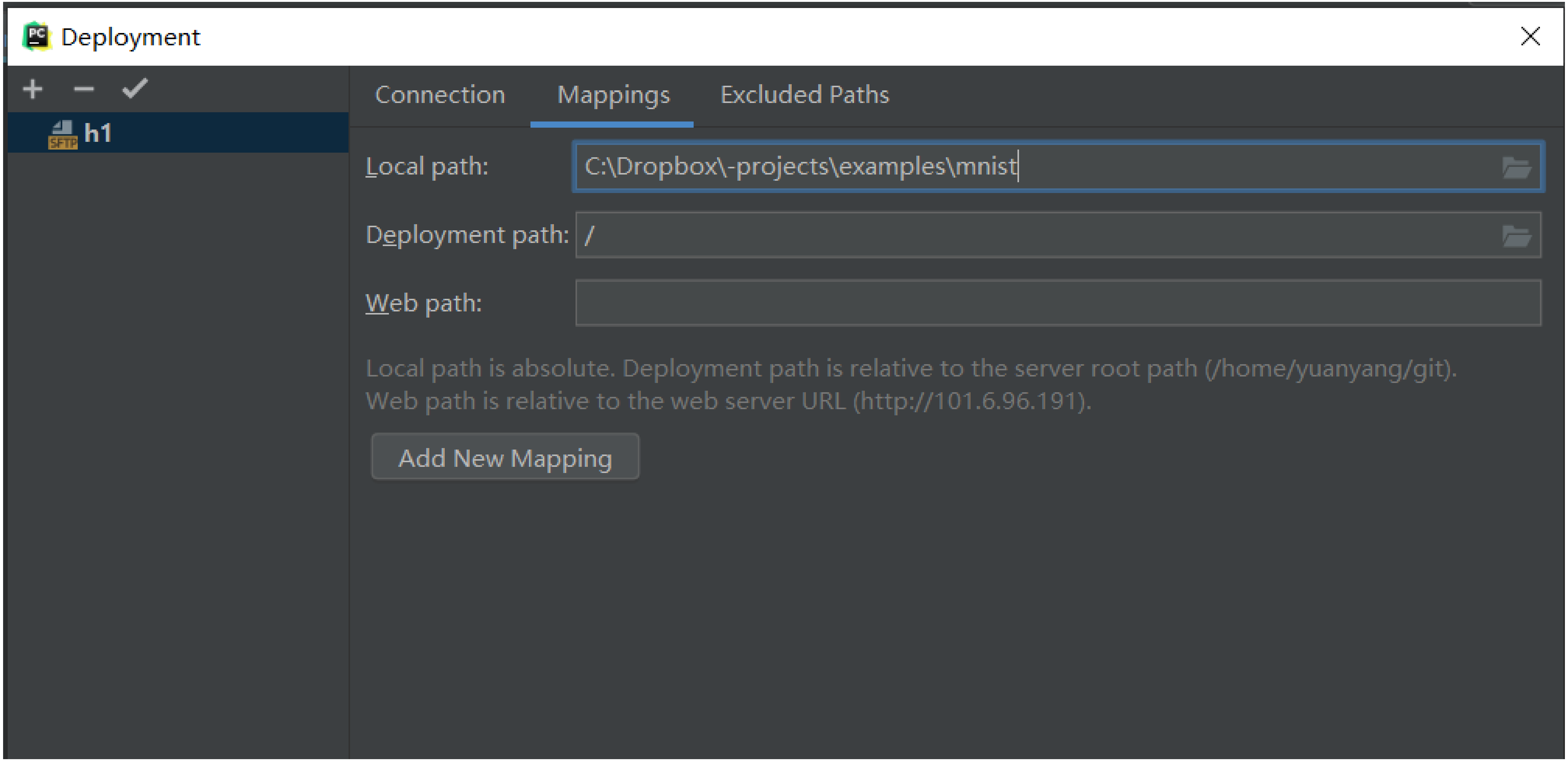

mnist [C:\Dropbox\-projects\examples\mnist] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mnist
Add Configuration...
Git:
Project
mnist C:\Dropbox\-projects\examples\mnis
main.py
README.md
requirements.txt
External Libraries
Scratches and Consoles
1: Project
Tasks & Contexts
IDE Scripting Console
Analyze Stack Trace...
Capture Memory Snapshot
Python Console...
Create setup.py
Show Code Coverage Data Ctrl+Alt+F6
Vim Emulator Ctrl+Alt+V
R Console...
Refresh R Package Index
Deployment
Add new Bash console
HTTP Client
Start SSH session...
Vagrant
Open CProfile snapshot
Upload to Default Server
Upload to... Ctrl+Alt+Shift+X
Download from Default Server
Download from...
Compare Local File with Deployed Version
Compare with Deployed to ...
Sync Local Subtree with Deployed
Sync with Deployed to ...
Configuration...
Options...
Automatic Upload
Browse Remote Host
SciView
R Packages
R Graphics
Database

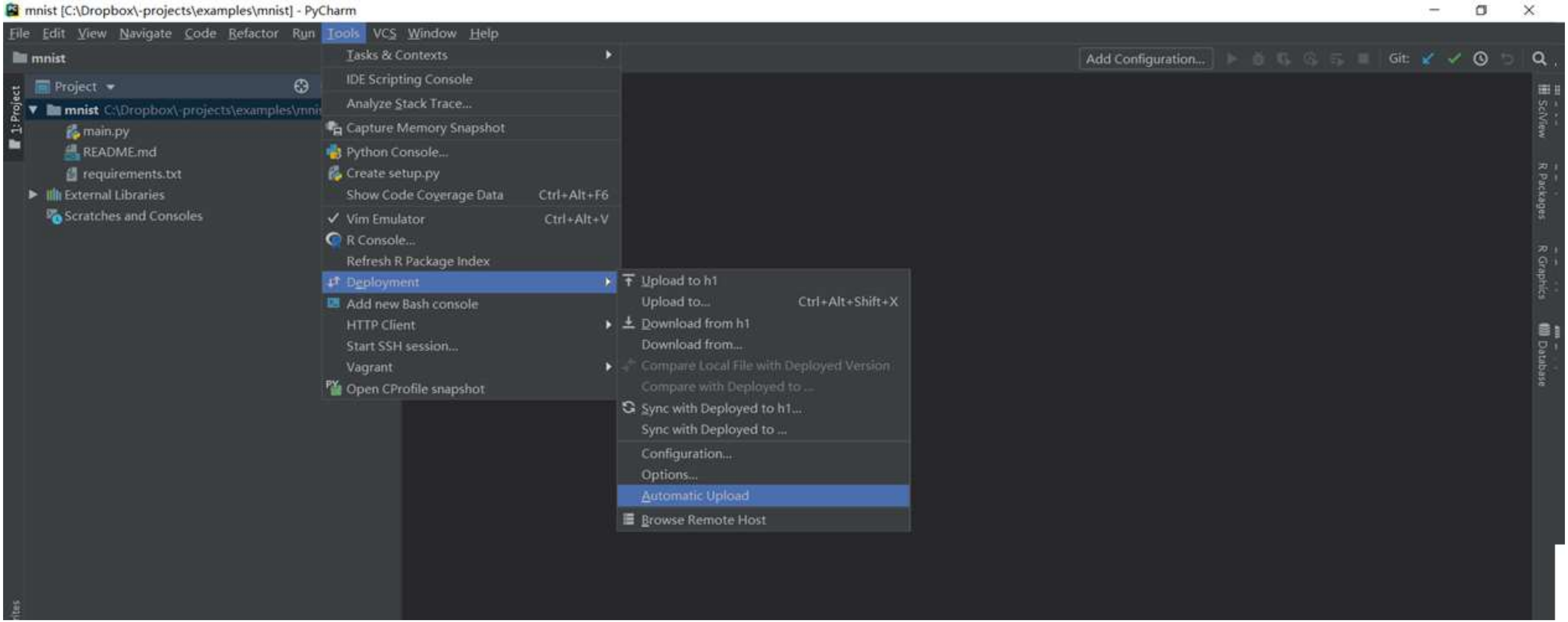
Use pytorch

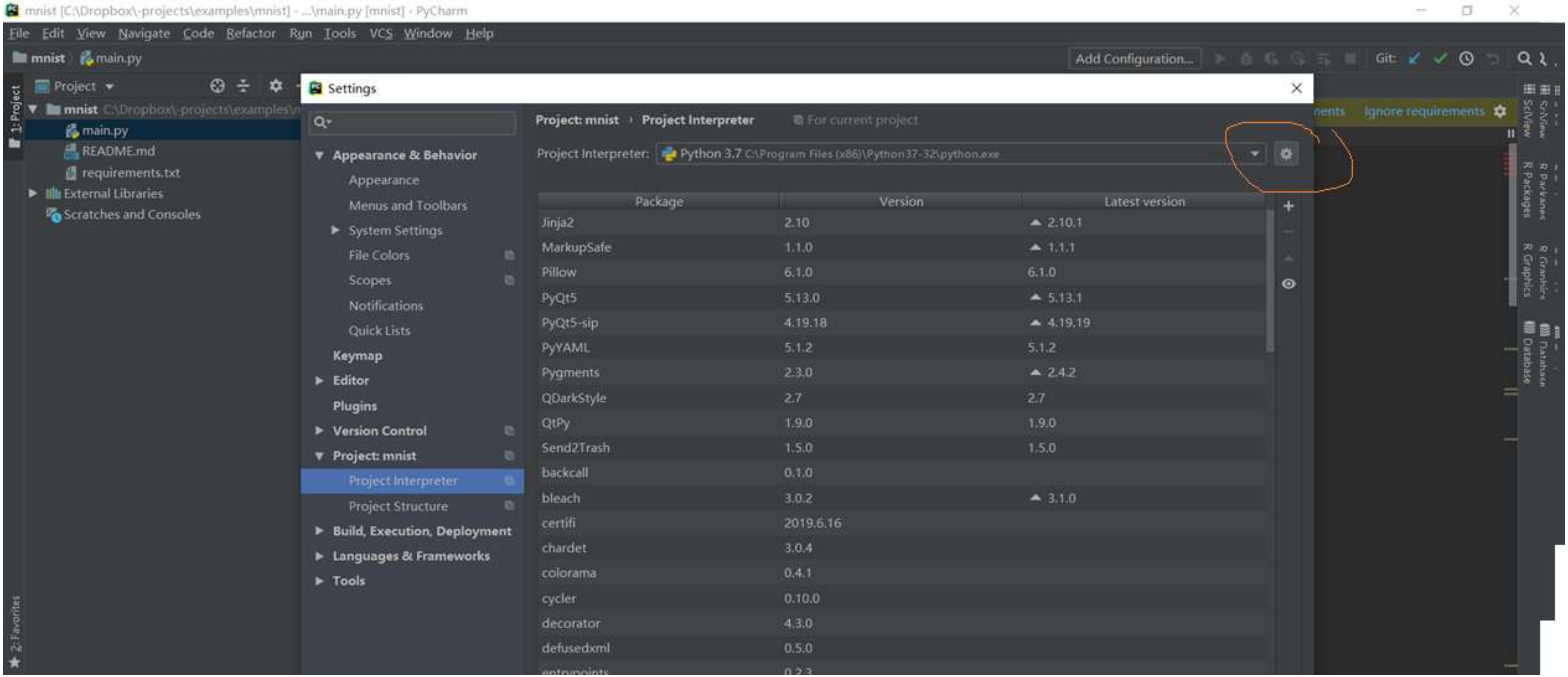

mnist [C:\Dropbox\-projects\examples\mnist] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mnist
Add Configuration...
Git:
Project
mnist C:\Dropbox\-projects\examples\mnist
main.py
README.md
requirements.txt
External Libraries
Scratches and Consoles
Deployment
FTP
FTPS
SFTP
Local or mounted folder
In place
Server group
Not configured
Please add a web server to configure
SciView
R Packages
R Graphics
Database
1: Project

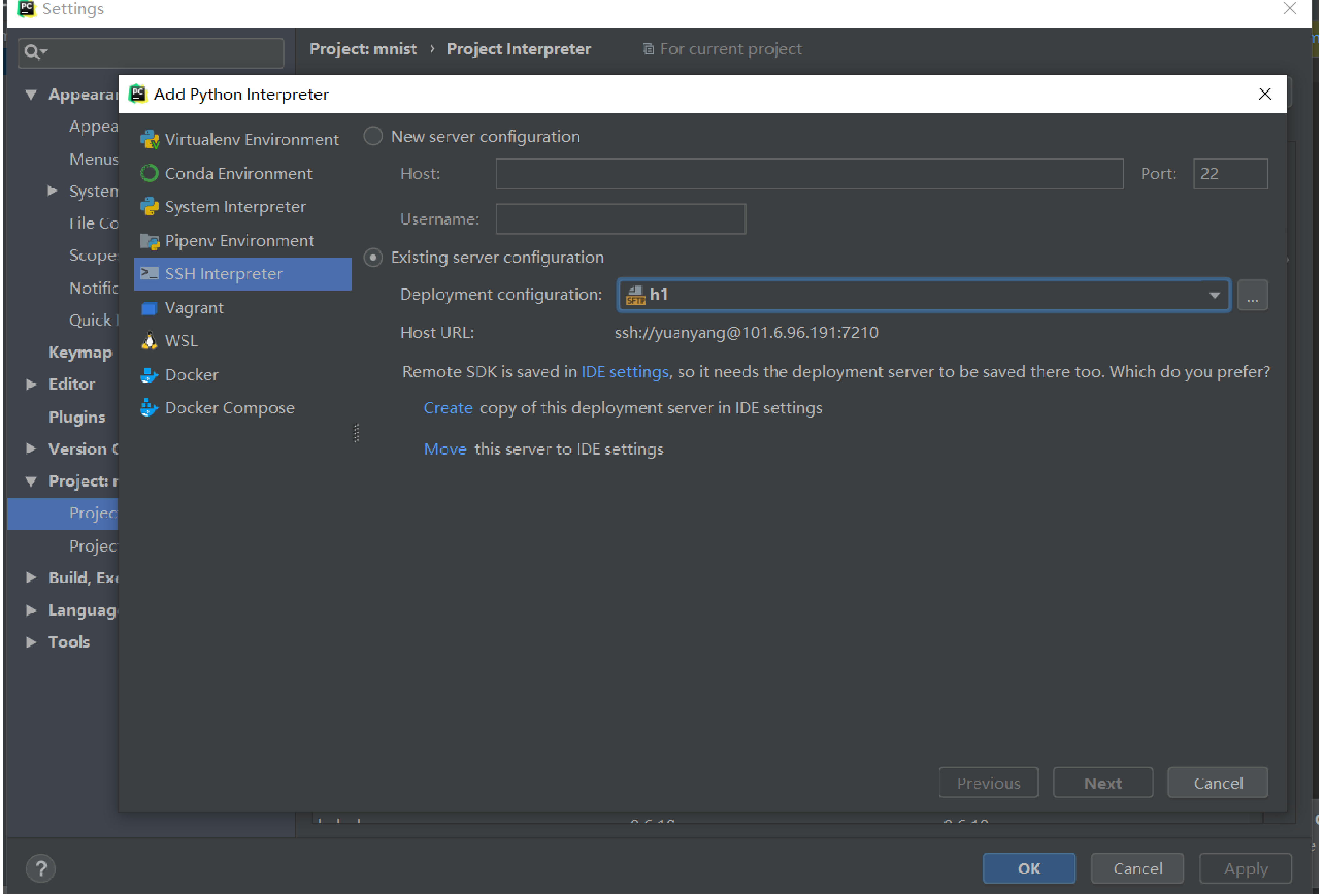


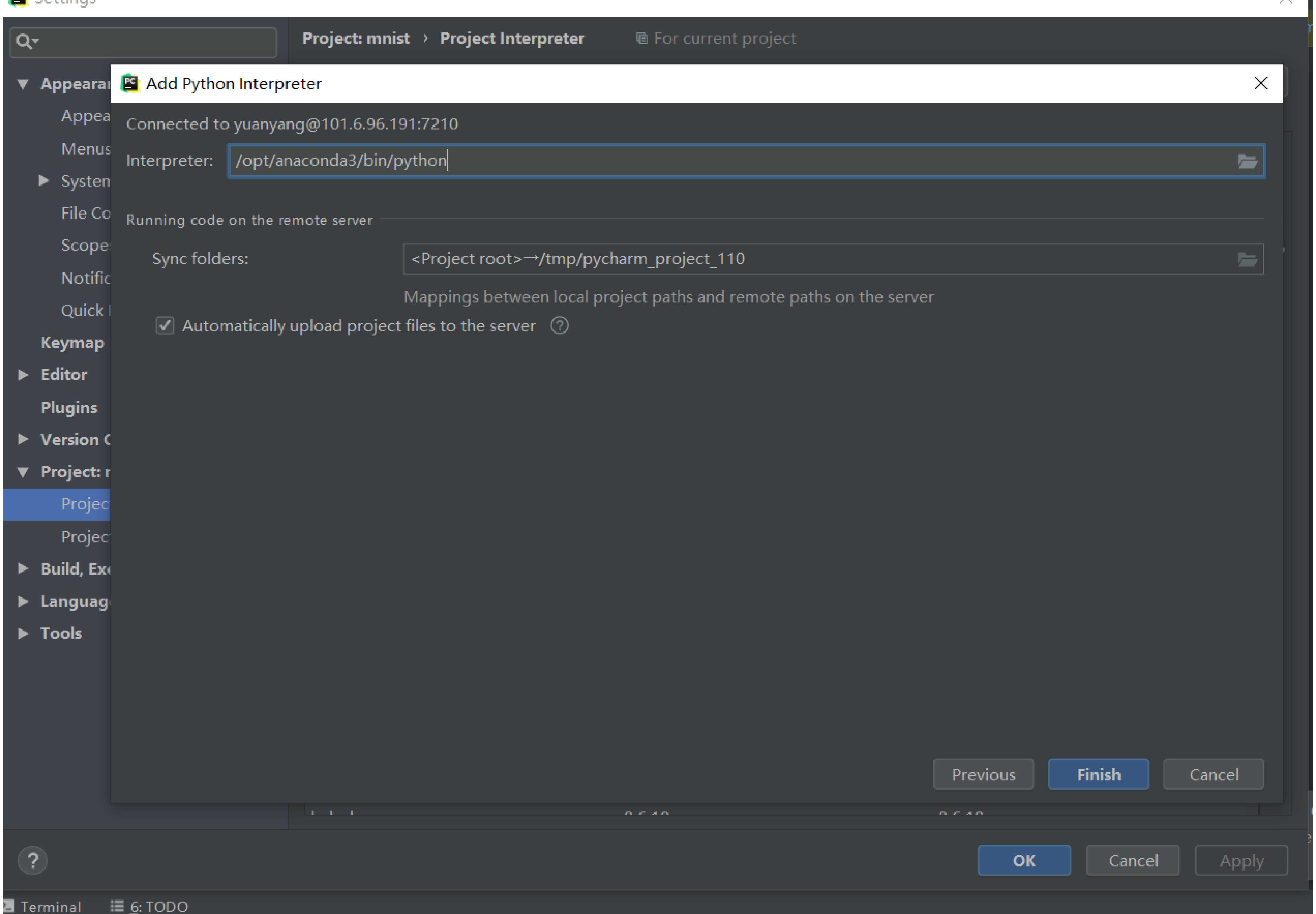
Deployment
h1
Connection
Mappings
Excluded Paths
Visible only for this project
Type:
SFTP
Host:
101.6.96.191
Port:
6201
User name:
yuanyang
Authentication:
Key pair OpenSSH or PuTTY
Private key path:
C:\Dropbox\yy\zm\new_high_private
Passphrase:
Save passphrase
Test Connection
Root path:
/
Autodetect
Web server URL:
http://101.6.96.191
Advanced
OK
Cancel


Deployment
h1
SFTP
Connection
Mappings
Excluded Paths
Local path:
C:\Dropbox\-projects\examples\mnist
Deployment path:
/
Web path:
Local path is absolute. Deployment path is relative to the server root path (/home/yuanyang/git).
Web path is relative to the web server URL (http://101.6.96.191).
Add New Mapping


mnist [C:\Dropbox\-projects\examples\mnist] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mnist
Project
mnist C:\Dropbox\-projects\examples\mnist
main.py
README.md
requirements.txt
External Libraries
Scratches and Consoles
Tasks & Contexts
IDE Scripting Console
Analyze Stack Trace...
Capture Memory Snapshot
Python Console...
Create setup.py
Show Code Coverage Data Ctrl+Alt+F6
Vim Emulator Ctrl+Alt+V
R Console...
Refresh R Package Index
Deployment
Add new Bash console
HTTP Client
Start SSH session...
Vagrant
Open CProfile snapshot
Upload to h1
Upload to... Ctrl+Alt+Shift+X
Download from h1
Download from...
Compare Local File with Deployed Version
Compare with Deployed to ...
Sync with Deployed to h1...
Sync with Deployed to ...
Configuration...
Options...
Automatic Upload
Browse Remote Host
Add Configuration...
Git:


mnist [C:\Dropbox\-projects\examples\mnist] - ...\main.py [mnist] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
mnist main.py
Add Configuration...
Git:
Project
mnist C:\Dropbox\-projects\examples\
main.py
README.md
requirements.txt
External Libraries
Scratches and Consoles
Settings
Appearance & Behavior
Appearance
Menus and Toolbars
System Settings
File Colors
Scopes
Notifications
Quick Lists
Keymap
Editor
Plugins
Version Control
Project: mnist
Project Interpreter
Project Structure
Build, Execution, Deployment
Languages & Frameworks
Tools
Project: mnist › Project Interpreter
For current project
Project Interpreter: Python 3.7 C:\Program Files (x86)\Python37-32\python.exe
Package Version Latest version
Jinja2 2.10 2.10.1
MarkupSafe 1.1.0 1.1.1
Pillow 6.1.0 6.1.0
PyQt5 5.13.0 5.13.1
PyQt5-sip 4.19.18 4.19.19
PyYAML 5.1.2 5.1.2
Pygments 2.3.0 2.4.2
QDarkStyle 2.7 2.7
QtPy 1.9.0 1.9.0
Send2Trash 1.5.0 1.5.0
backcall 0.1.0
bleach 3.0.2 3.1.0
certifi 2019.6.16
chardet 3.0.4
colorama 0.4.1
cycler 0.10.0
decorator 4.3.0
defusedxml 0.5.0
Ignore requirements
SciView
R Packages
R Graphics
Database
2: Favorites
1: Project

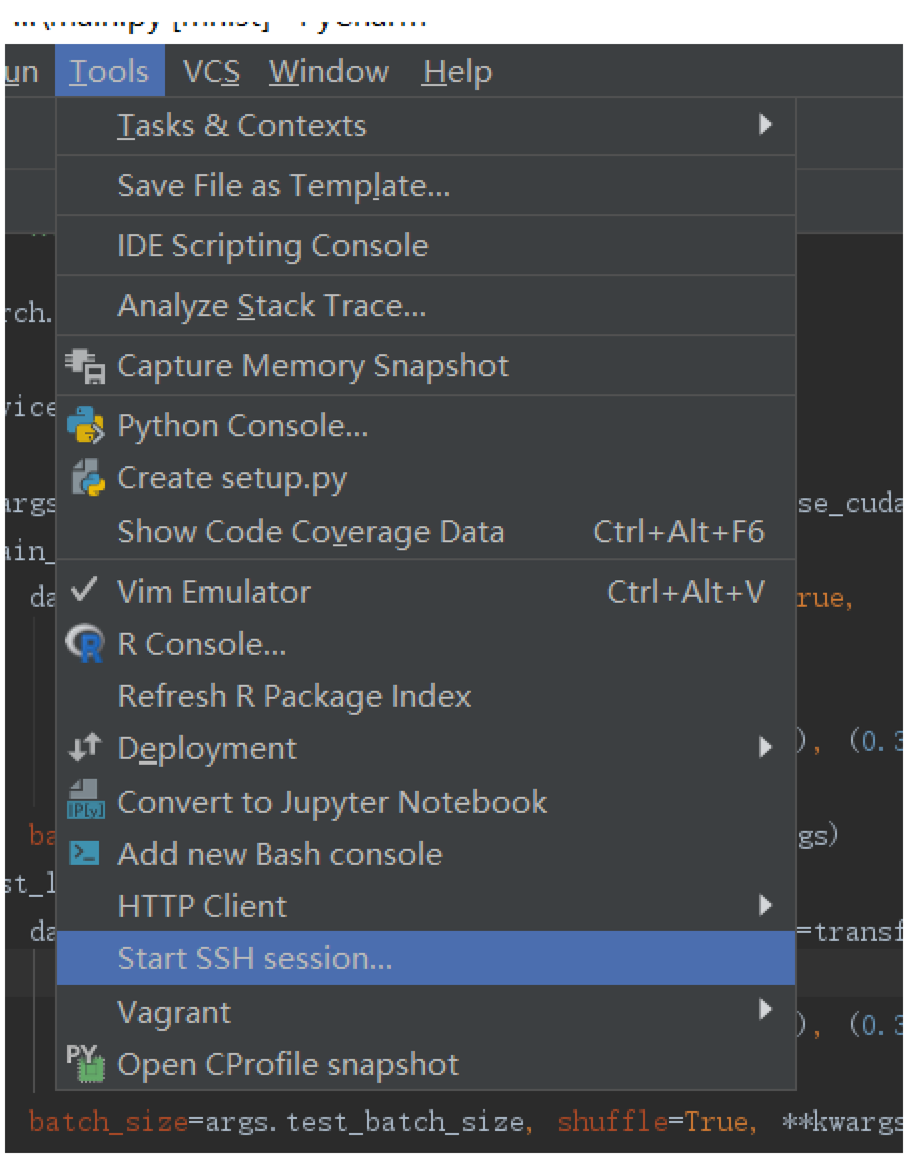

Settings
Project: mnist › Project Interpreter
For current project
Add Python Interpreter
Virtualenv Environment
Conda Environment
System Interpreter
Pipenv Environment
SSH Interpreter
Vagrant
WSL
Docker
Docker Compose
New server configuration
Host:
Port: 22
Username:
Existing server configuration
Deployment configuration: h1
Host URL: ssh://yuanyang@101.6.96.191:7210
Remote SDK is saved in IDE settings, so it needs the deployment server to be saved there too. Which do you prefer?
Create copy of this deployment server in IDE settings
Move this server to IDE settings
Previous
Next
Cancel
Appearance
Keymap
Editor
Plugins
Tools
OK
Cancel
Apply



Project: mnist › Project Interpreter
For current project
Add Python Interpreter
Connected to yuanyang@101.6.96.191:7210
Interpreter:
/opt/anaconda3/bin/python
Running code on the remote server
Sync folders:
<Project root>→/tmp/pycharm_project_110
Mappings between local project paths and remote paths on the server
Automatically upload project files to the server
Previous
Finish
Cancel
Appearan
Appea
Menus
System
File Co
Scope
Notific
Quick
Keymap
Editor
Plugins
Version C
Project: r
Projec
Projec
Build, Exe
Languag
Tools
OK
Cancel
Apply
Terminal
6: TODO

Now you can run& update

- Run the code directly (suggested by Mingkuan Xu): shift+F10
- Run the code through SSH (most common way)
- Run the code using Pycharm SSH (convenient inside pycharm)

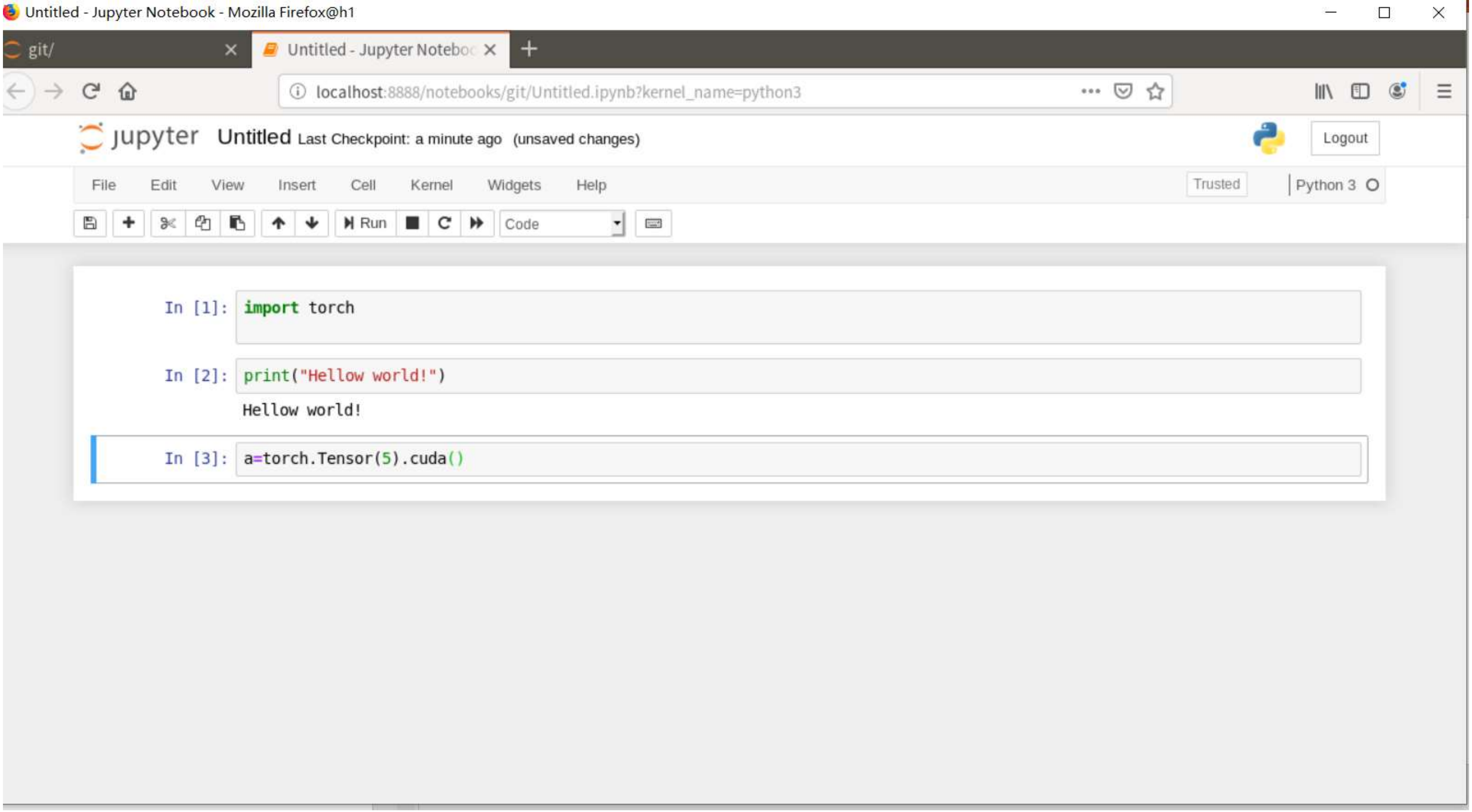


Use jupyter to run code

```
Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Sat Sep 28 22:45:40 2019 from 123.114.52.117
(base) yuanyang@h1:~$ jupyter notebook
[I 22:46:38.025 NotebookApp] JupyterLab extension loaded from /opt/anaconda3/lib/python3.7/site-packages/jupyterlab
[I 22:46:38.025 NotebookApp] JupyterLab application directory is /opt/anaconda3/share/jupyter/lab
[I 22:46:38.026 NotebookApp] Serving notebooks from local directory: /home/yuanyang
[I 22:46:38.026 NotebookApp] The Jupyter Notebook is running at:
[I 22:46:38.026 NotebookApp] http://localhost:8888/?token=d1fafa4340eac9cc6211a7d966dde504f471637a91d24ba4
[I 22:46:38.026 NotebookApp]  or http://127.0.0.1:8888/?token=d1fafa4340eac9cc6211a7d966dde504f471637a91d24ba4
[I 22:46:38.026 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 22:46:46.479 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/yuanyang/.local/share/jupyter/runtime/nbserver-25142-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=d1fafa4340eac9cc6211a7d966dde504f471637a91d24ba4
     or http://127.0.0.1:8888/?token=d1fafa4340eac9cc6211a7d966dde504f471637a91d24ba4

[E 22:46:55.411 NotebookApp] Could not open static file ''
[W 22:46:55.457 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (127.0.0.1) 10.08ms referer=http://localhost:8888/tree?toke
ac9cc6211a7d966dde504f471637a91d24ba4
[W 22:46:55.468 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (127.0.0.1) 1.11ms referer=http://localhost:8888/tree?token
c9cc6211a7d966dde504f471637a91d24ba4
[I 22:47:03.495 NotebookApp] Creating new notebook in /git
[I 22:47:03.519 NotebookApp] Writing notebook-signing key to /home/yuanyang/.local/share/jupyter/notebook_secret
[W 22:47:04.097 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (127.0.0.1) 1.38ms referer=http://localhost:8888/notebooks/
ipynb?kernel_name=python3
[W 22:47:04.283 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (127.0.0.1) 2.63ms referer=http://localhost:8888/notebooks/
ipynb?kernel_name=python3
[I 22:47:05.046 NotebookApp] Kernel started: 1a6423b2-2a99-4ff1-ad5a-feeacb079268
[I 22:47:05.420 NotebookApp] Adapting from protocol version 5.1 (kernel 1a6423b2-2a99-4ff1-ad5a-feeacb079268) to 5.3 (client).
```


Untitled - Jupyter Notebook - Mozilla Firefox@h1
git/
Untitled - Jupyter Notebo
localhost:8888/notebooks/git/Untitled.ipynb?kernel_name=python3
jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [1]: import torch
In [2]: print("Hellow world!")
Hellow world!
In [3]: a=torch.Tensor(5).cuda()

Model

- Init:
  - **Define** the network components
  - Super.__init__()
  - Convolutional layers
  - Linear layers
- Forward:
  - **Run** the network
  - x=F.relu(self.conv1(x))
  - What is F?
  - What is relu, max_pool2d?
  - What is view, log_softmax?

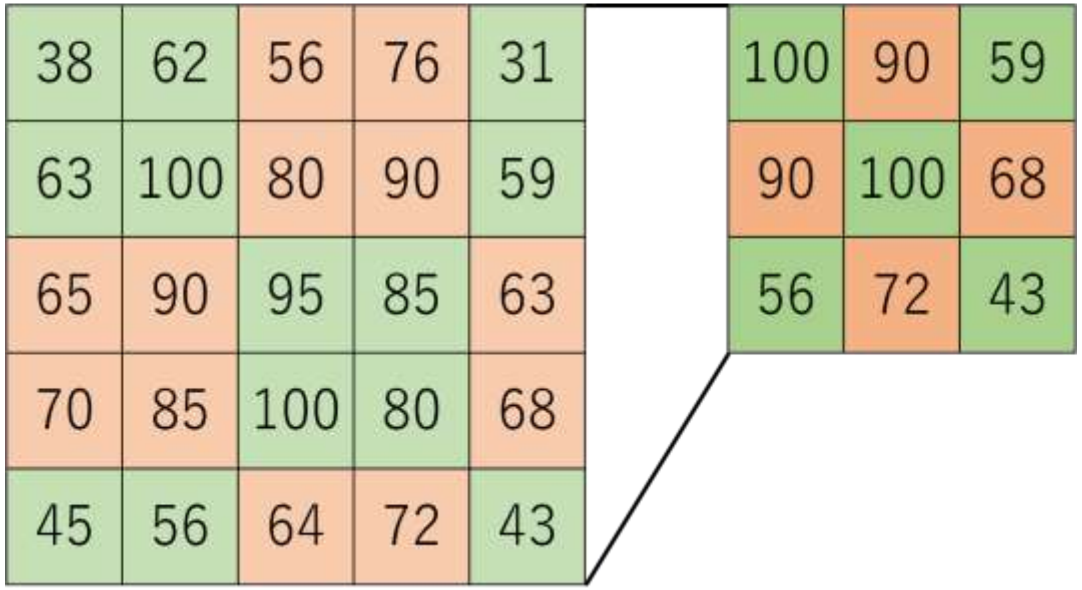
```python
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 20, 5, 1)
        self.conv2 = nn.Conv2d(20, 50, 5, 1)
        self.fc1 = nn.Linear(4*4*50, 500)
        self.fc2 = nn.Linear(500, 10)


    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.max_pool2d(x, 2, 2)
        x = F.relu(self.conv2(x))
        x = F.max_pool2d(x, 2, 2)
        x = x.view(-1, 4*4*50)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return F.log_softmax(x, dim=1)
```

Model

- Relu: activation calculation
  - Relu(x)=0 if $x < 0$

    =x if $x \geq 0$
- Max_pool2d



- Relu: activation calculation
  - Relu(x)=0 if $x < 0$

    =x if $x \geq 0$
- Max_pool2d

Model

- view
  - Reshape the output
  - -1 means by calculation
  - E.g., A size=100*100
  - A.view(-1,200)=50*200
- Log_softmax
  - Convert vectors into probabilities
  - Take log

```python
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 20, 5, 1)
        self.conv2 = nn.Conv2d(20, 50, 5, 1)
        self.fc1 = nn.Linear(4*4*50, 500)
        self.fc2 = nn.Linear(500, 10)



    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.max_pool2d(x, 2, 2)
        x = F.relu(self.conv2(x))
        x = F.max_pool2d(x, 2, 2)
        x = x.view(-1, 4*4*50)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return F.log_softmax(x, dim=1)
```

Model

- More questions: check https://pytorch.org/docs/stable/index.html
- Also check the tutorial https://pytorch.org/tutorials/

log_softmax

`torch.nn.functional.log_softmax(input, dim=None, _stacklevel=3, dtype=None)` [SOURCE]

Applies a softmax followed by a logarithm.

While mathematically equivalent to log(softmax(x)), doing these two operations separately is slower, and numerically unstable. This function uses an alternative formulation to compute the output and gradient correctly.

See `LogSoftmax` for more details.

Parameters

- **input** (*Tensor*) – input
- **dim** (*int*) – A dimension along which log_softmax will be computed.
- **dtype** (`torch.dtype`, optional) – the desired data type of returned tensor. If specified, the input tensor is casted to `dtype` before the operation is performed. This is useful for preventing data type overflows. Default: None.