

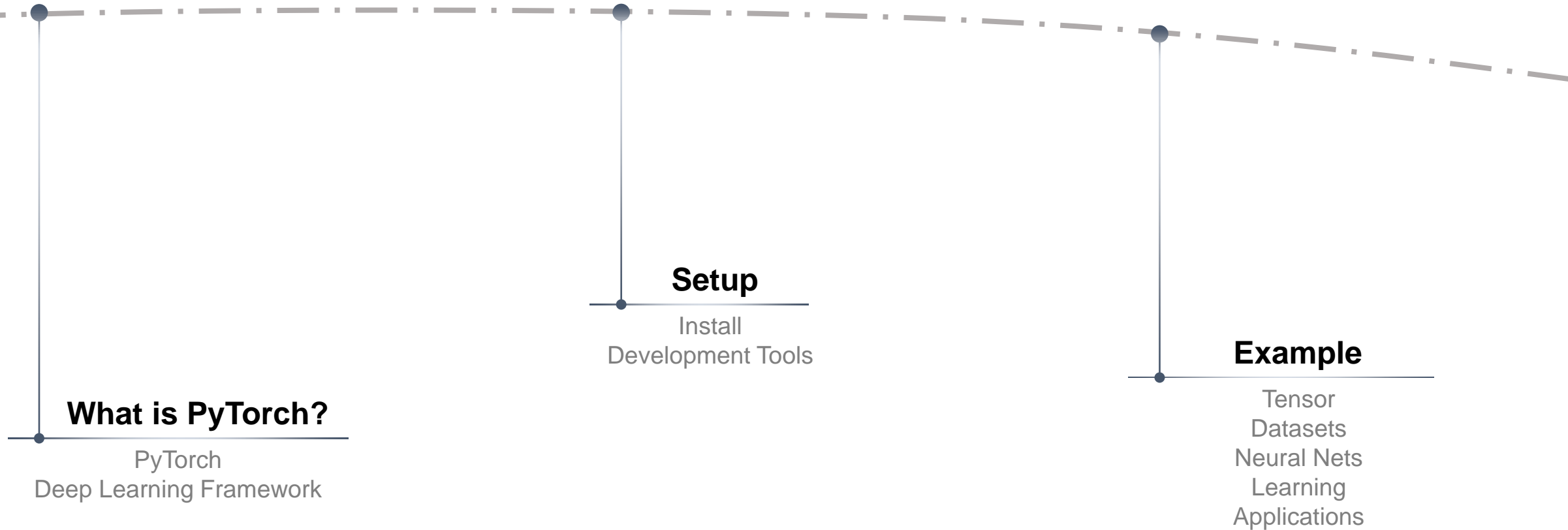
# Touch to PyTorch

: From basic to vanilla GAN

*ISL Lab Seminar*

*Hansol Kang*

# Contents



# What is PyTorch?

- PyTorch

# PYTORCH

an open source **machine learning library for Python**.  
It is primarily developed by **Facebook's artificial-intelligence research group**.



Vs.

# TensorFlow

 Google AI

# What is PyTorch?

- Deep Learning Framework

**theano**  
 Nov. 2010  
 Written in : Python  
 Interface : Python

**Caffe**  
 Dec. 2013  
 Written in : C++  
 Interface : Python, MATLAB, C++

**torch**  
 Jul. 2014  
 Written in : C, Lua  
 Interface : C, Lua

<p><b>Keras</b> **                  Mar. 2015                  Written in : Python                  Interface : Python, R</p>	<p><b>TensorFlow</b>                  Nov. 2015                  Written in : C++, Python, CUDA                  Interface : Python, C/C++, Java, Go, R, Julia</p>	<p><b>PYTORCH</b>                  Oct. 2016                  Written in : Python, C, CUDA                  Interface : Python</p>
---	--	--

Recommend to choose these framework

**Caffe2**  
 Apr. 2017  
 Written in :  
 Interface : Python, C++

DL4J(Java)  
 Chainer(Python)  
 MXNet(C++, Python, Julia, MATLAB, JavaScript, Go, R, Scala, Perl)  
 CNTK(Python, C++),  
 TF Learn(Python)  
 TF-Slim(Python)  
 Etc.

Cf. "What is this? Gum? It's GAN.", pp. 21-22

# Setup

- Install

<https://www.anaconda.com/download/>

## Anaconda 5.2 For Windows Installer

Python 3.6 version \*

↓ Download

[64-Bit Graphical Installer \(631 MB\)](#) ?

[32-Bit Graphical Installer \(506 MB\)](#)

or

Python 2.7 version \*

↓ Download

[64-Bit Graphical Installer \(564 MB\)](#) ?

[32-Bit Graphical Installer \(443 MB\)](#)

# Setup

- Install

<https://pytorch.org/>

Select your preferences and run the install command. Please ensure that you are on the latest pip and numpy packages. Anaconda is our recommended package manager. You can also **install previous versions of PyTorch**. Note that LibTorch is only available for C++.

PyTorch Build	Stable		Preview		
Your OS	Linux		Mac		Windows
Package	Conda	Pip		LibTorch	Source
Language	Python 2.7	Python 3.5	Python 3.6	Python 3.7	C++
CUDA	8.0	9.0		9.2	None
Run this Command:	<pre>conda install pytorch -c pytorch pip3 install torchvision</pre>				

C++ : Preview,  
Linux only

# Setup

- Install

```
conda create -n PyTorch python=3.6
activate PyTorch
conda install pytorch cuda90 -c pytorch
pip install torchvision
```

- **conda create**는 환경 생성하는 명령어. PyTorch 뿐만 아니라 Tensorflow 같은 다른 딥러닝 프레임워크를 사용한다거나 하위 파이썬 버전도 사용해야하는 경우 환경마다 설정해주면, 디펜던시가 꼬이지 않음.
- **-n 환경명, python=파이썬버전** 입력. 환경설정 리스트는 **conda env list**를 입력하면 확인 가능.
- **activate**는 해당 환경을 활성화 시키는 명령어. 반대로 환경을 빠져나오는 명령어는 **deactivate**.
- PyTorch를 설치하는 명령어는 **conda install pytorch cuda90 -c pytorch..**
- **torchvision**은 딥러닝 학습에 많이 사용되는 데이터셋, 네트워크 구조, 이미지 변환과 같은 기능을 제공하므로 설치하는 것을 권장.

# Setup

- Development Tools



: Intellisense

: Cell based execution



: Extension program

: Intellisense  
 : Cell based execution  
 : Management of python env.



Visual Studio

: Startup file

: Intellisense  
 : Management of python env.  
 : GitHub  
 : AI tool package



VS Code

: Intellisense  
 : Environment setting

: Insane extension program



# Setup

- Development Tools

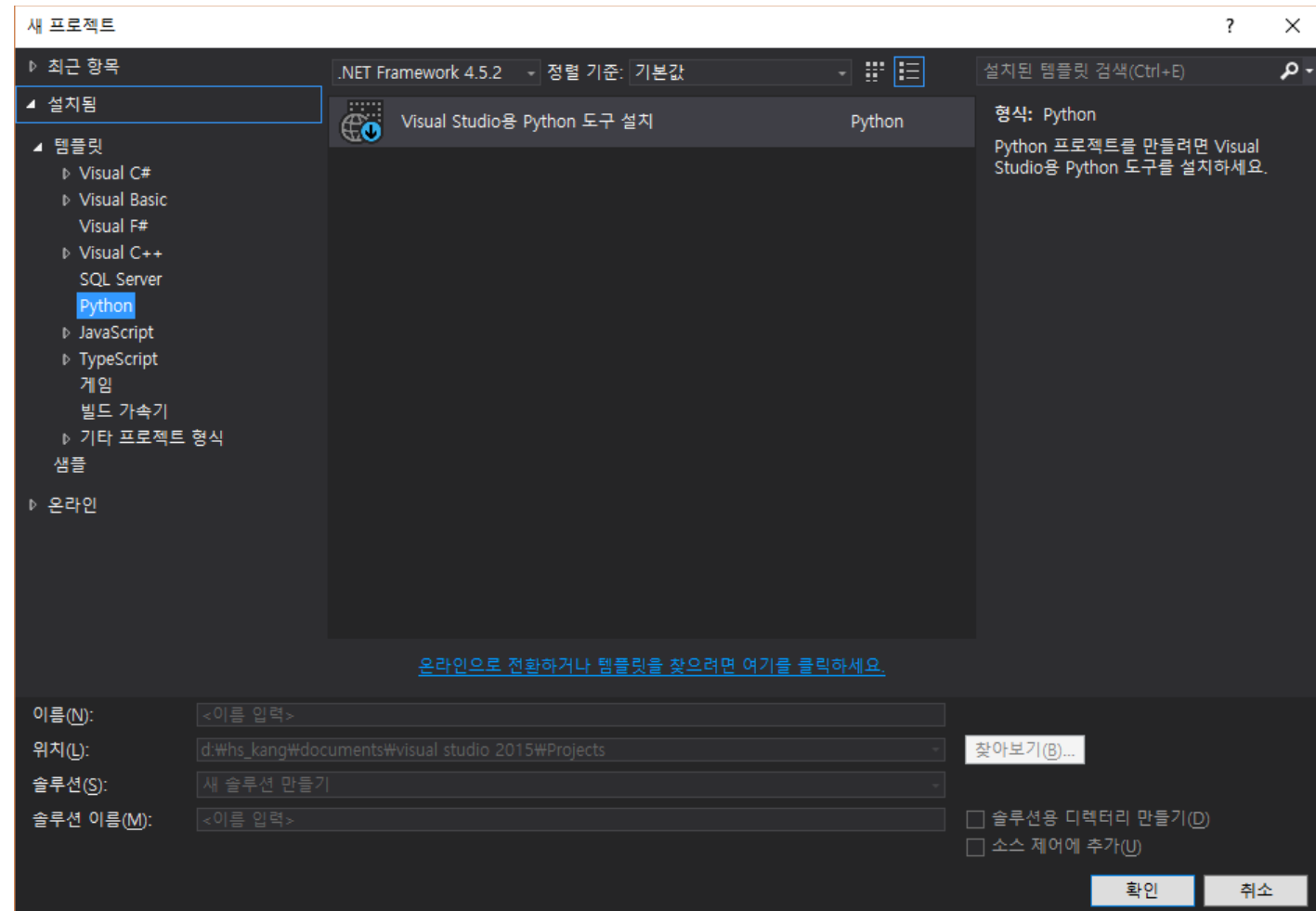
Personal opinion :



# Setup

- Development Tools – Visual Studio

## File-New-Project

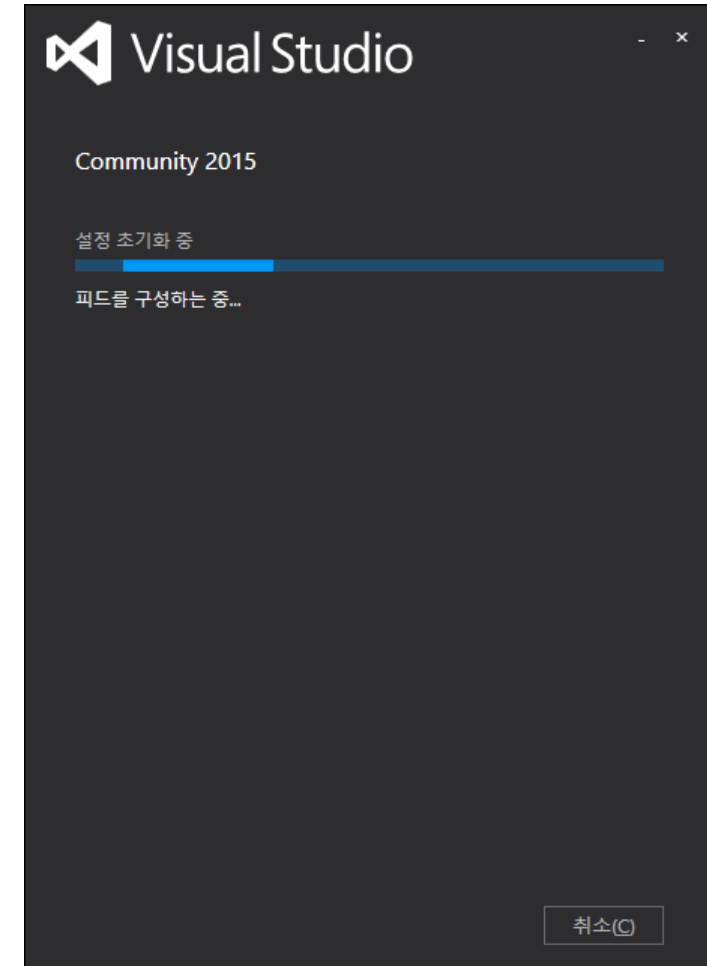
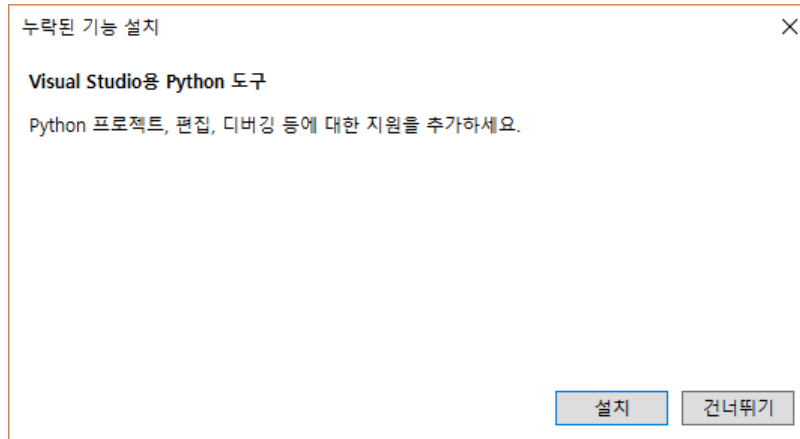


# Setup

- Development Tools – Visual Studio

File-New-Project

Python 도구 설치



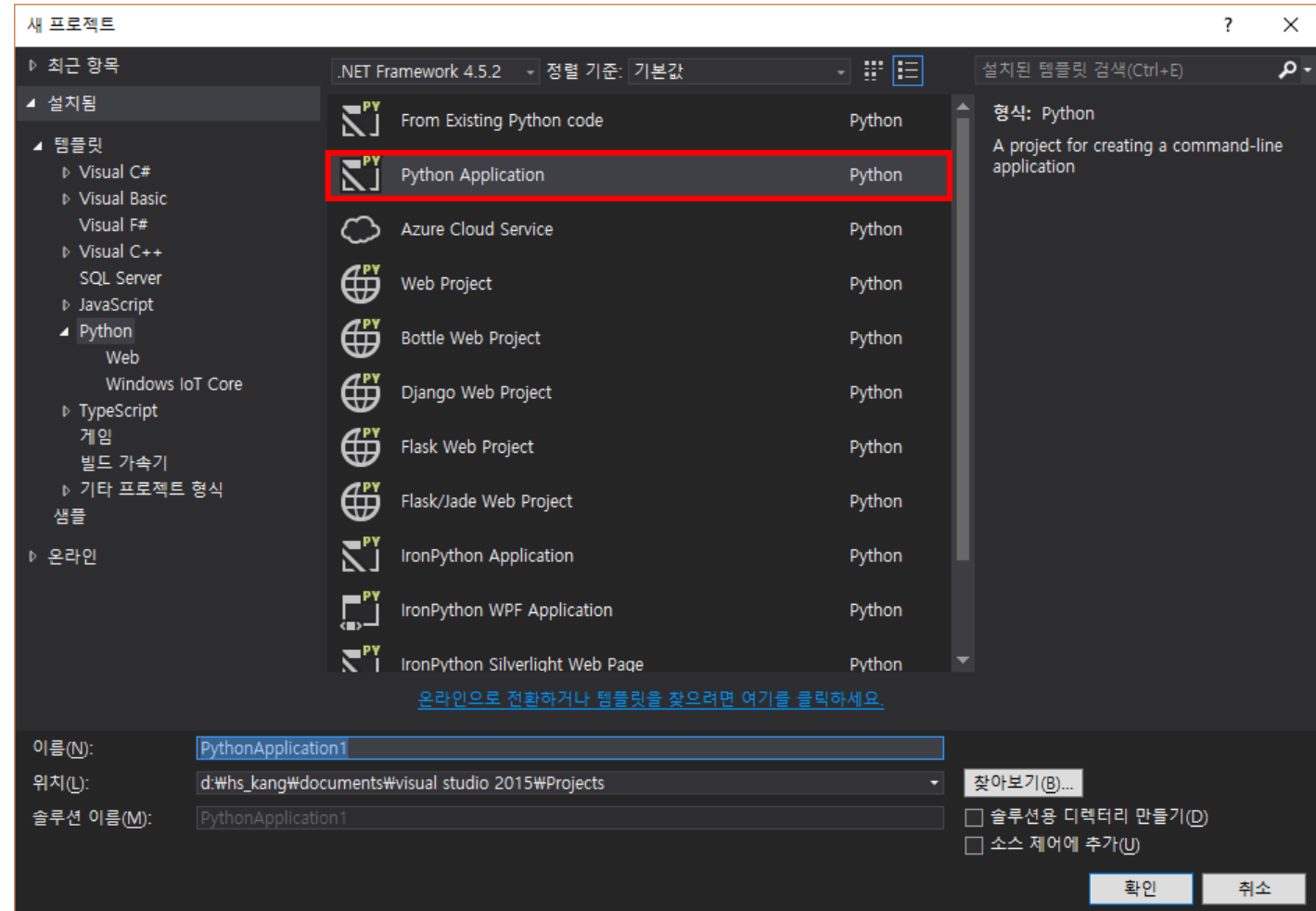
# Setup

- Development Tools – Visual Studio

File-New-Project

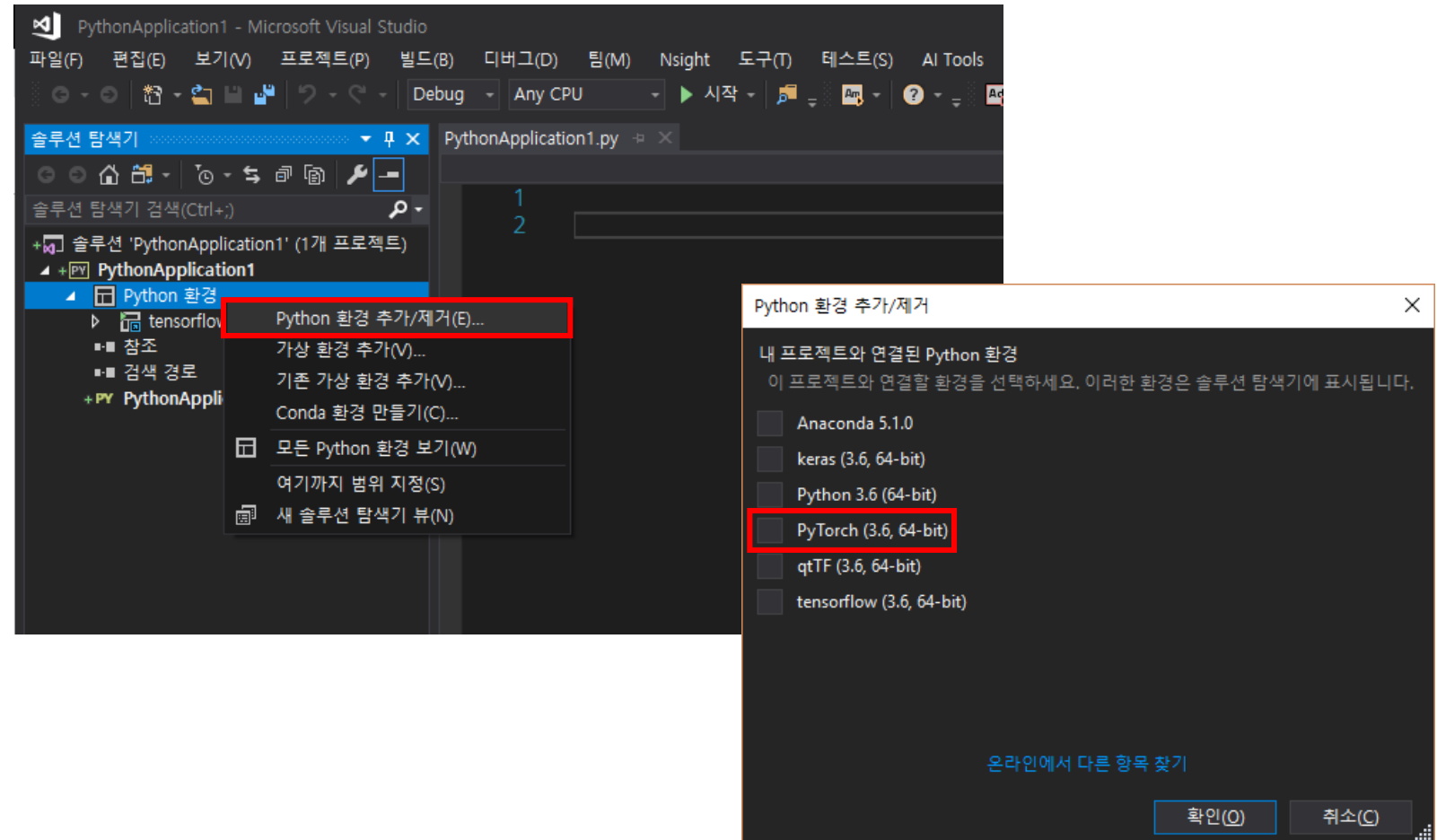
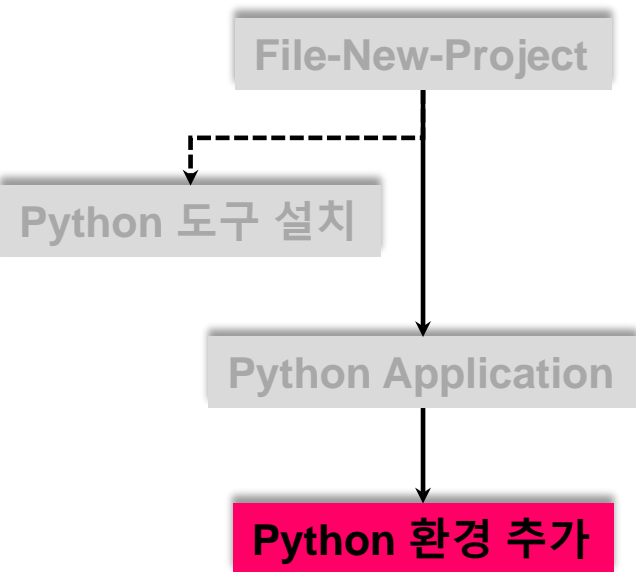
Python 도구 설치

Python Application



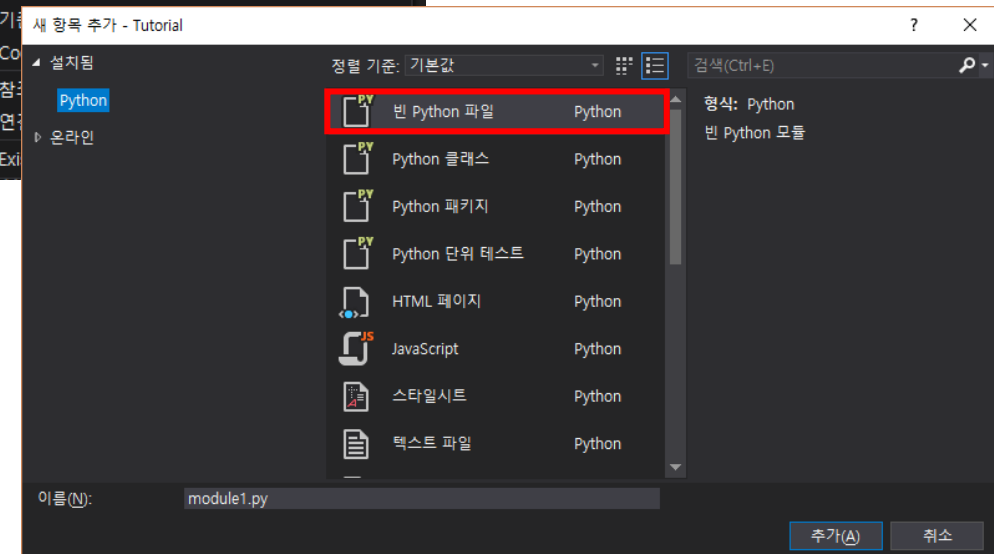
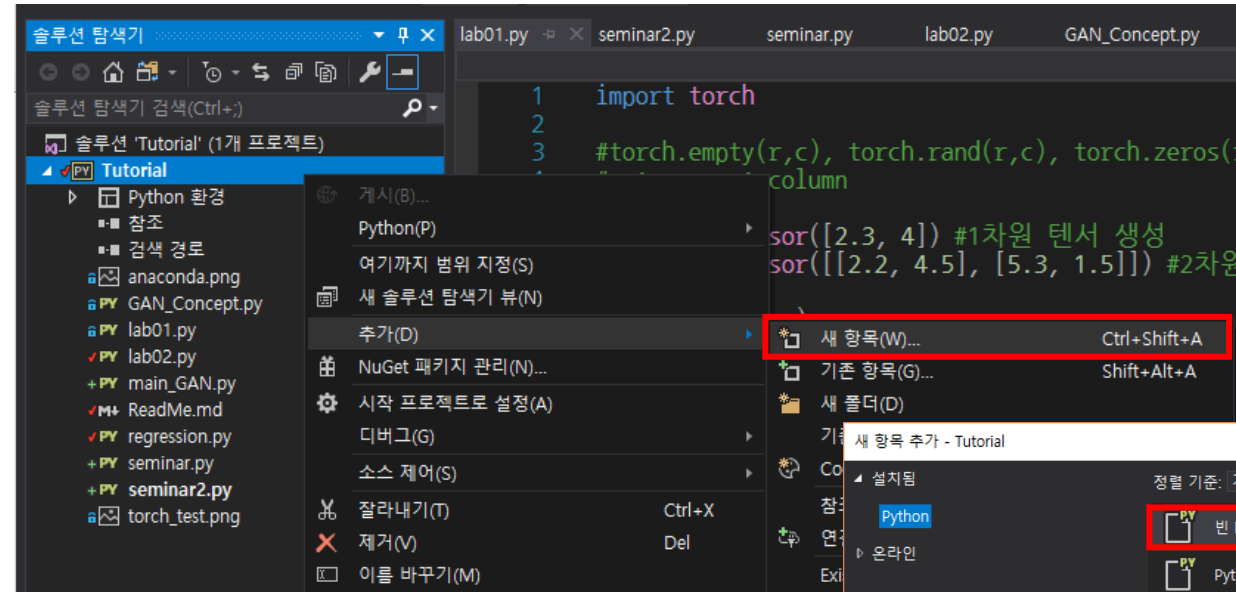
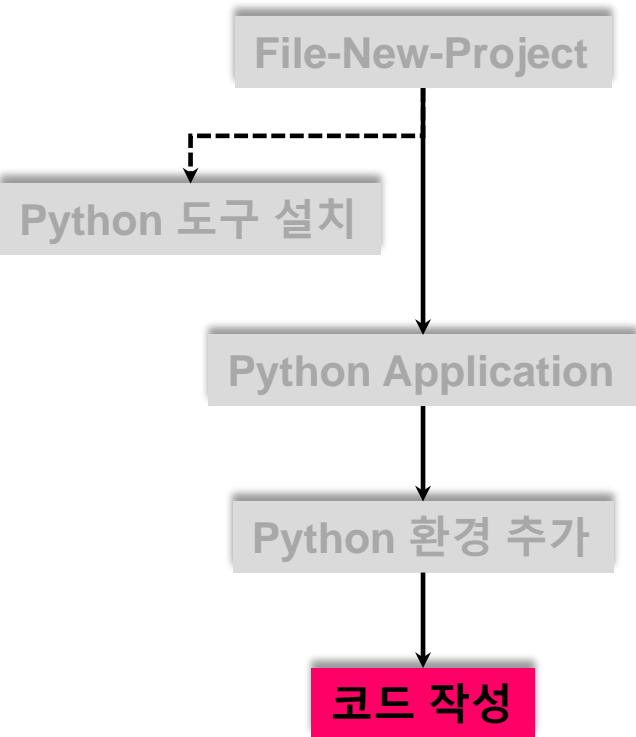
# Setup

- Development Tools – Visual Studio



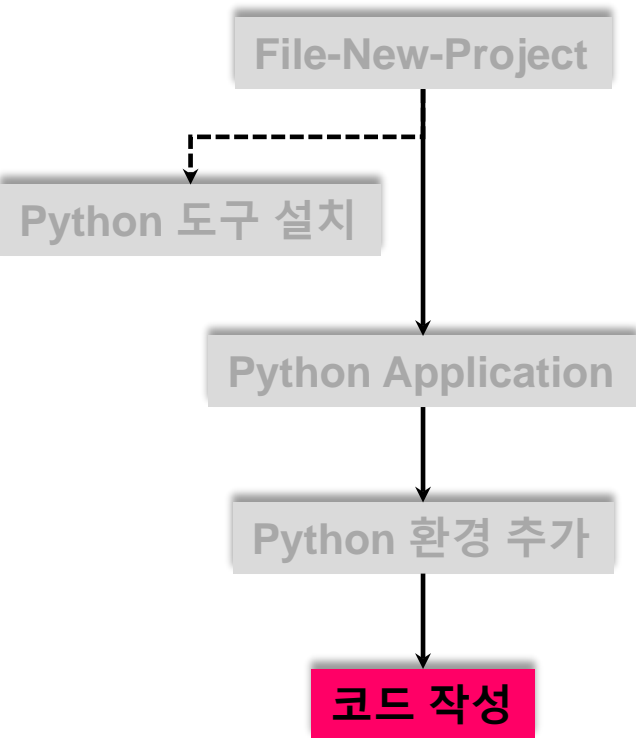
# Setup

- Development Tools – Visual Studio



# Setup

- Development Tools – Visual Studio

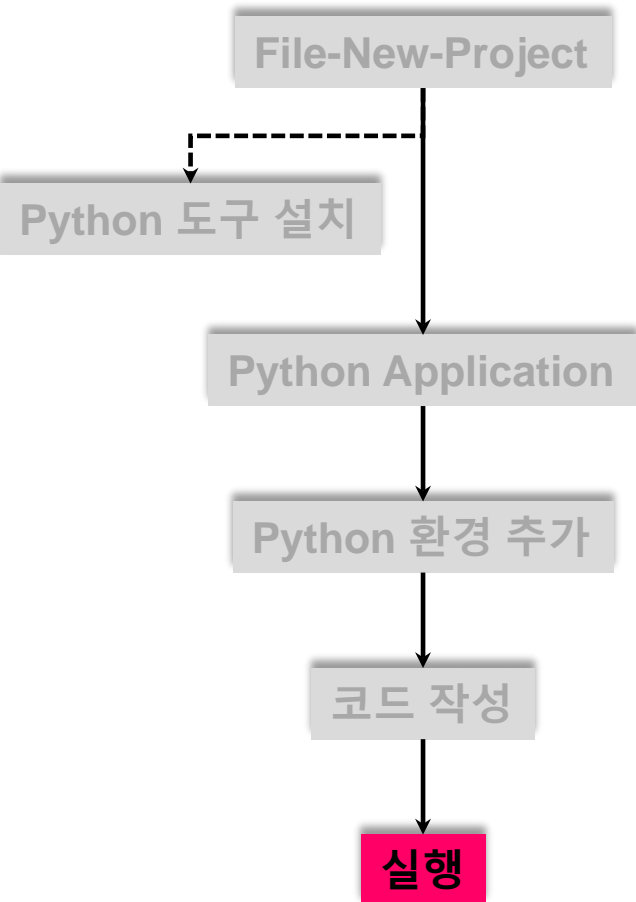


```

lab01.py seminar2.py seminar.py lab02.py GAN_Concept.py regression.py
1 import torch
2
3 #torch.empty(r,c), torch.rand(r,c), torch.zeros(r,c,dtype)
4 #r : row, c: column
5
6 a = torch.tensor([2.3, 4]) #1차원 텐서 생성
7 b = torch.tensor([[2.2, 4.5], [5.3, 1.5]]) #2차원 텐서 생성
8
9 print("a : ", a)
10 print("size of a : ", a.size(), "\n")
11
12 print("b : ", b)
13 print("size of b : ", b.size(), "\n")
14
15 x = a.new_ones(5,3,dtype=torch.double) #텐서 재사용
16 print("x : ", x)
17
18 #텐서를 넘겨 받음. dtype override
19 x = torch.randn_like(x, dtype=torch.float)
20 print("print : ", x)
21
22 #y.add_(x), 언더바 붙이면 in-place 자기 자신
23 c = torch.rand(2,2)
24 c.add_(b)
25 print(c)
  
```

# Setup

- Development Tools – Visual Studio



```

C:\WINDOWS\system32\cmd.exe
a : tensor([ 2.3000,  4.0000])
size of a : torch.Size([2])

b : tensor([[ 2.2000,  4.5000],
            [ 5.3000,  1.5000]])
size of b : torch.Size([2, 2])

x : tensor([[ 1.,  1.,  1.],
            [ 1.,  1.,  1.],
            [ 1.,  1.,  1.],
            [ 1.,  1.,  1.]], dtype=torch.float64)
print : tensor([[ -0.8485,  0.1334, -0.9038],
                [ 0.9339,  0.1457, -0.6499],
                [-0.9843,  0.2999,  2.0542],
                [ 0.0576,  0.6289,  1.3946],
                [-1.3774, -0.0733,  1.0720]])
tensor([[ 2.6828,  5.2298],
        [ 5.7302,  2.3466]])
torch.Size([4, 4]) torch.Size([16]) torch.Size([2, 8])
x: tensor([[ 0.2956,  0.9266, -0.0140,  0.1061],
           [-0.6696, -0.8309,  0.2865, -0.2927],
           [ 0.3135, -0.5125, -0.4398, -1.3477],
           [-0.7555, -0.8231, -0.8202,  1.1952]])
  
```

: 시작 파일로 설정-> Ctrl+F5



# Setup

- Development Tools – PyCharm

JET BRAINS

Tools Languages Solutions Support Store

PyCharm Coming in 2018.2 What's New Features Docs & Demos Buy

Download

Click

PC **PyCharm**

Python IDE  
for Professional Developers

DOWNLOAD NOW

Full-fledged Professional or Free Community

WHY PYCHARM

Cookies and IP addresses allow us to deliver and improve our web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

-----

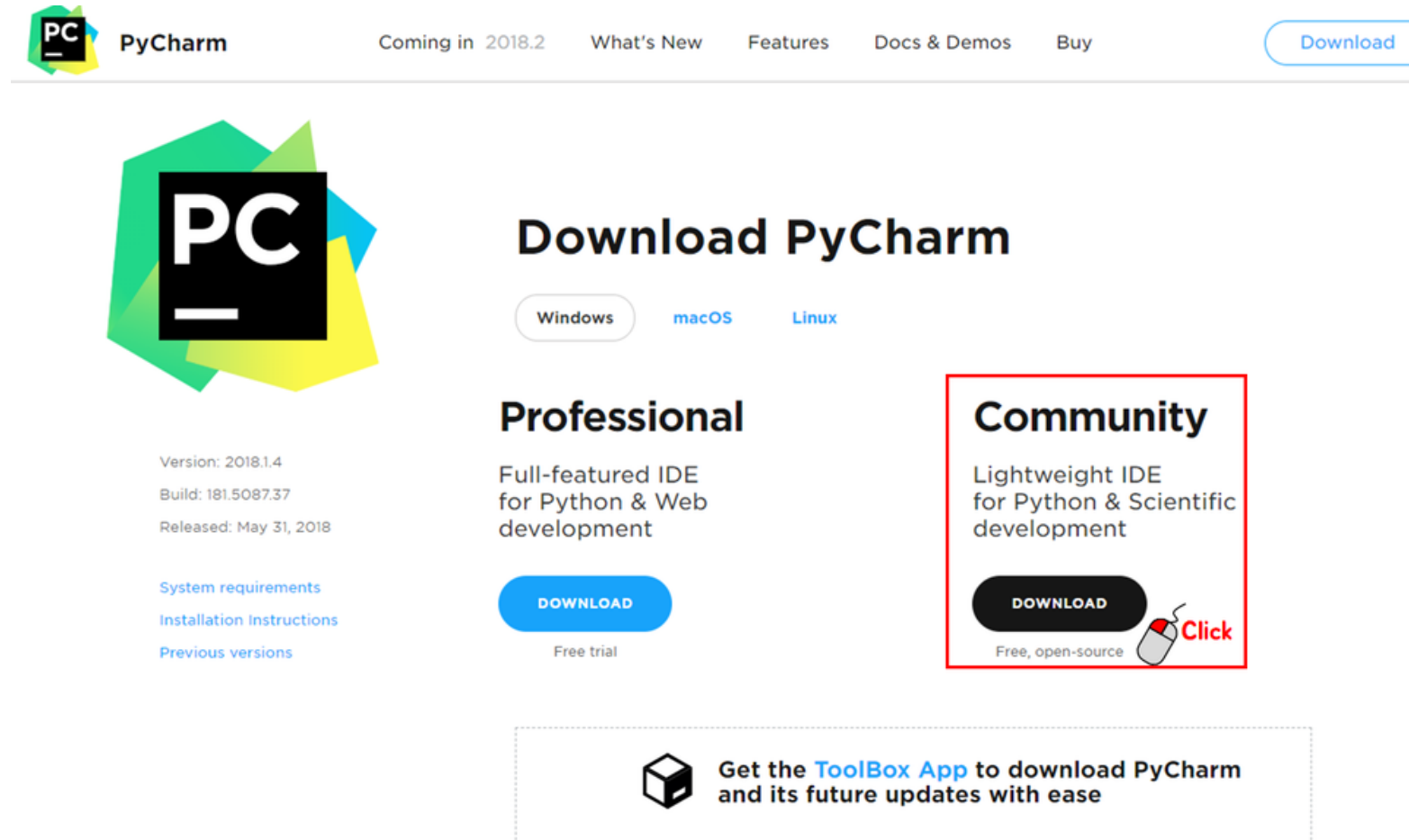
| JetBrains may use cookies and my IP address to collect individual statistics and to provide me with personalized offers and ads subject to the [Privacy Policy](#) and the [Terms of Use](#). JetBrains may use [third-party services](#) for this purpose. I can revoke my consent at any time by visiting the [Do Not Sell My Info](#) link.

| [Yes, I agree] [No, thanks]

-----

# Setup

- Development Tools – PyCharm



The screenshot shows the PyCharm website's download page. At the top, there is a navigation bar with the PyCharm logo, the text 'PyCharm', and links for 'Coming in 2018.2', 'What's New', 'Features', 'Docs & Demos', and 'Buy'. A 'Download' button is located in the top right corner. Below the navigation bar, the main content area features a large PyCharm logo on the left. To the right of the logo, the heading 'Download PyCharm' is displayed, followed by three tabs: 'Windows' (selected), 'macOS', and 'Linux'. Below the tabs, there are two main product options: 'Professional' and 'Community'. The 'Professional' option is described as a 'Full-featured IDE for Python & Web development' and includes a blue 'DOWNLOAD' button with a 'Free trial' label below it. The 'Community' option is described as a 'Lightweight IDE for Python & Scientific development' and includes a black 'DOWNLOAD' button with a mouse cursor icon and the text 'Click' next to it, and 'Free, open-source' below. A red rectangular box highlights the 'Community' option. At the bottom of the page, a dashed-line box contains a cube icon and the text 'Get the ToolBox App to download PyCharm and its future updates with ease'.

PC PyCharm

Coming in 2018.2 What's New Features Docs & Demos Buy

Download

## Download PyCharm

Windows macOS Linux

### Professional

Full-featured IDE for Python & Web development

DOWNLOAD

Free trial

### Community

Lightweight IDE for Python & Scientific development

DOWNLOAD

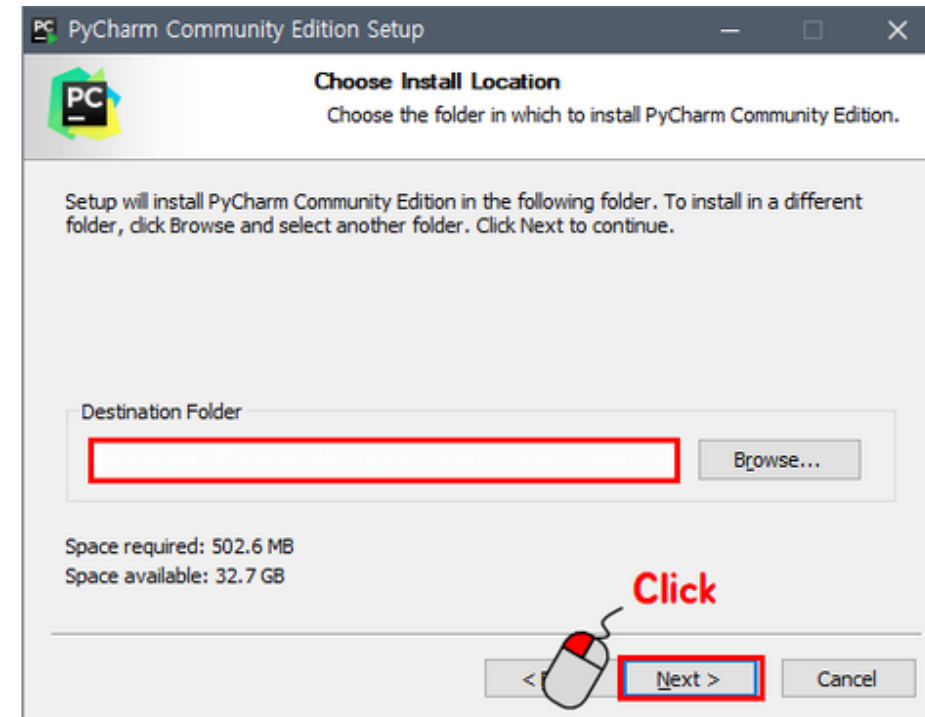
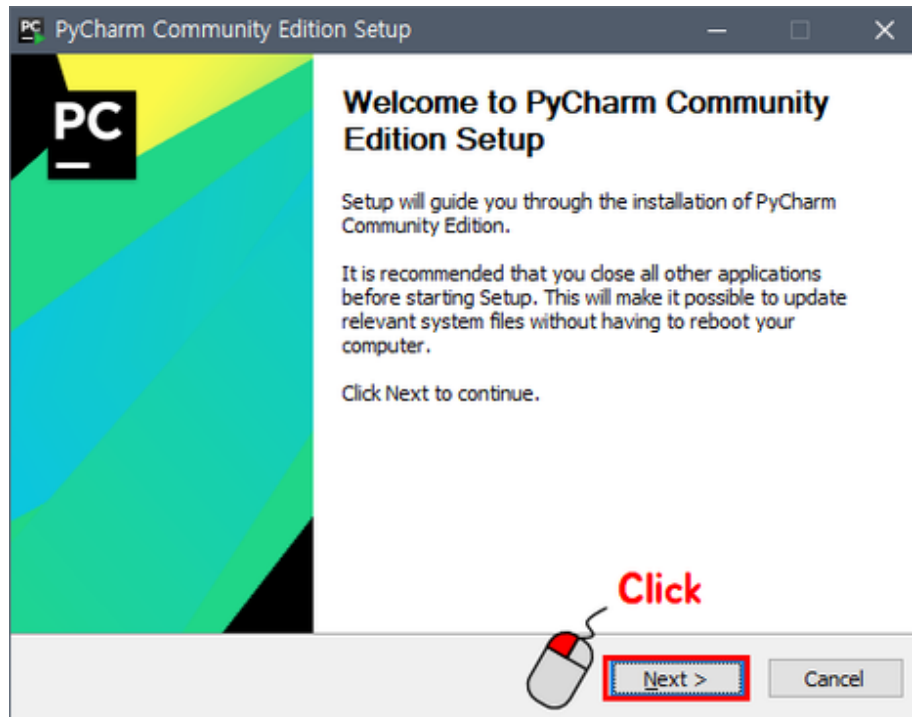
Click

Free, open-source

Get the [ToolBox App](#) to download PyCharm and its future updates with ease

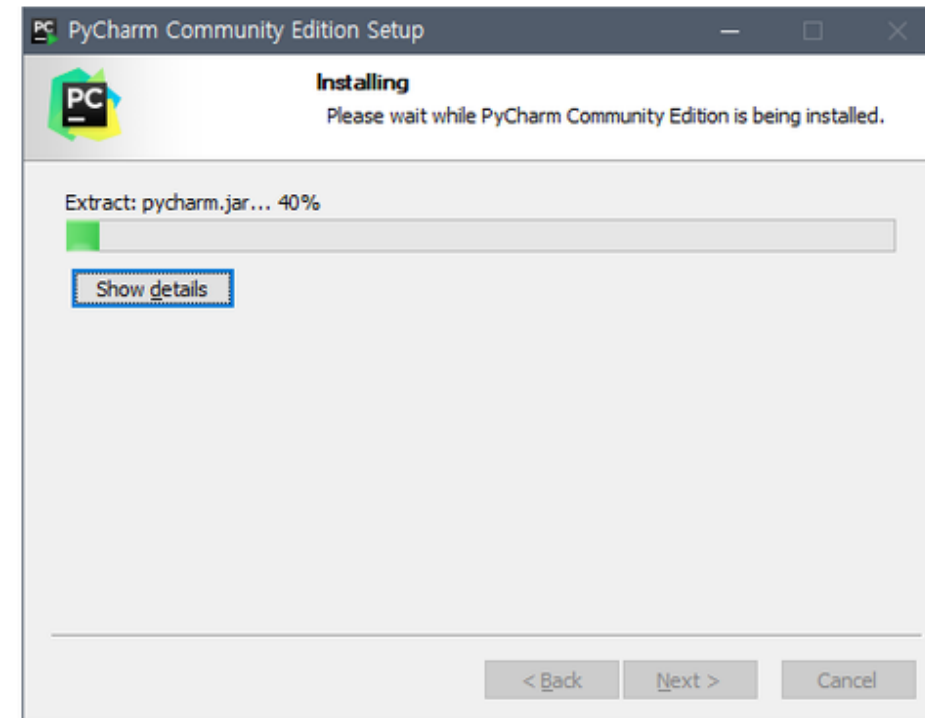
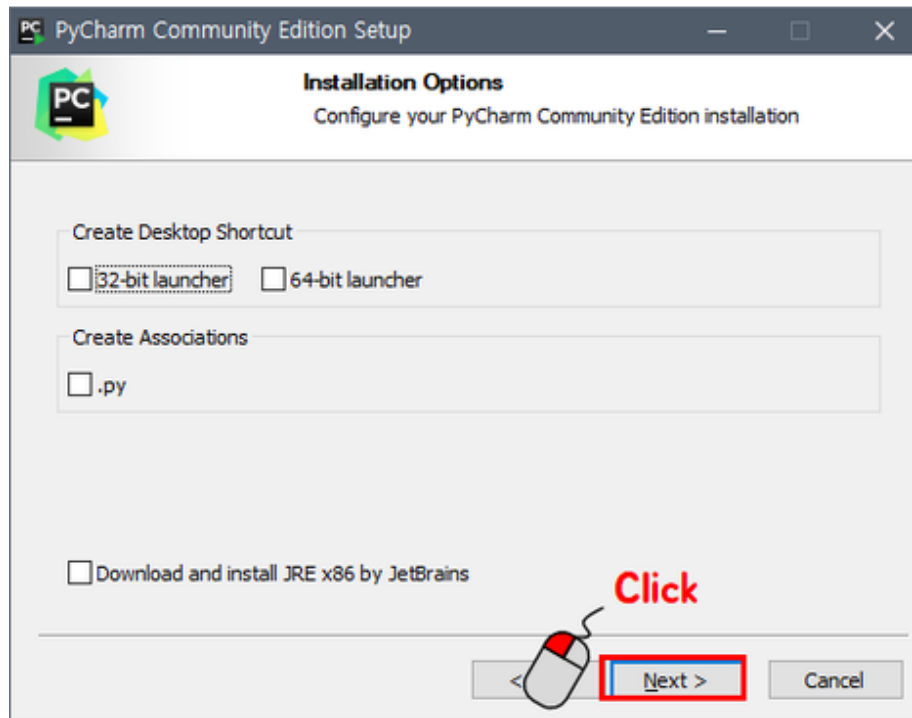
# Setup

- Development Tools – PyCharm



# Setup

- Development Tools – PyCharm



# Setup

- Development Tools – PyCharm

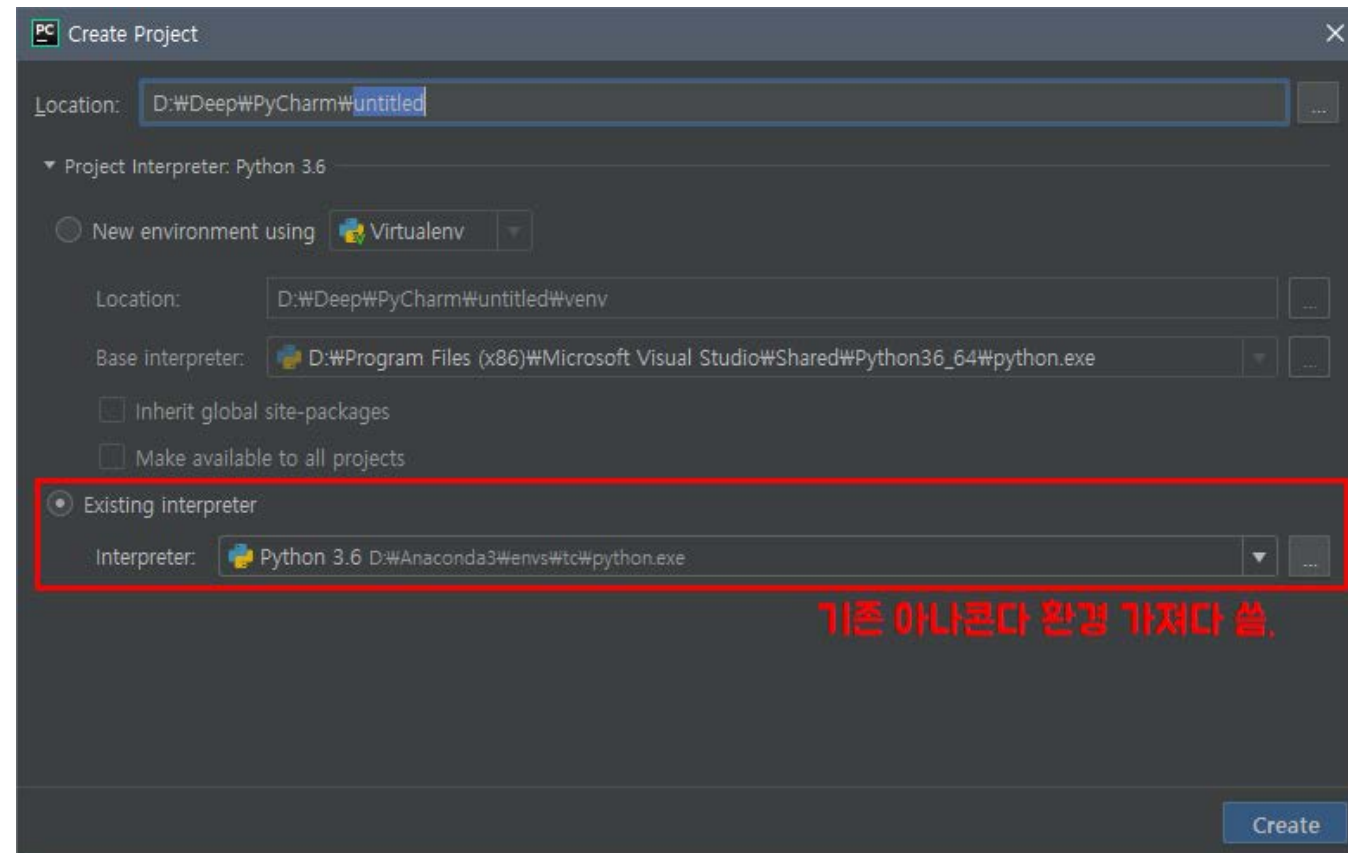
The image illustrates the steps to configure PyCharm settings:

- Open the **File** menu and select **Settings...** (highlighted with a red box and a mouse cursor icon labeled "Click").
- In the **Settings** dialog, navigate to **Editor** (highlighted with a red box) and select **Color Scheme** (highlighted with a red box). A dropdown menu shows the **Monokai** color scheme selected, with the Korean text "컬러 테마" (Color Theme) next to it.
- In the **Settings** dialog, navigate to **Keymap** (highlighted with a red box) and select **Visual Studio** in the dropdown menu (highlighted with a red box). The Korean text "단축키 설정" (Shortcut Key Setting) is visible in the background.

# Setup

- Development Tools – PyCharm

## File-New Project



# Setup

- Development Tools – PyCharm

File-New Project

코드 작성

The screenshot shows the PyCharm IDE interface. The 'File' menu is open, and the 'New' submenu is also open, with 'Python File' selected. The Project tool window on the left shows the project name 'Blog' highlighted with a red box. A red mouse cursor icon with the word 'Click' is positioned over the project name. Another red mouse cursor icon with the word 'Click' is positioned over the 'Python File' option in the 'New' submenu. A separate window titled 'Test01.py' shows the following code:

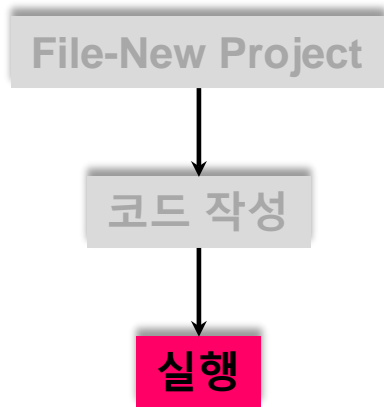
```

1 import numpy as np
2
3 a = np.array([1, 2, 3])
4 b = np.array([2, 3, 4])
5 c = a+b
6 print(c)

```

# Setup

- Development Tools – PyCharm



: Ctrl+F5

Cf. TODO

“Too easy, but important things”, pp. 42-43

The screenshot shows the PyCharm IDE interface. The top pane displays a Python script named 'Test01.py' with the following code:

```

1 import numpy as np
2
3 a = np.array([1, 2, 3])
4 b = np.array([2, 3, 4])
5 c = a+b
6 print(c)
7
8
  
```

The bottom pane shows the 'Run' console for 'Test01'. The output is:

```

D:\Anaconda3\envs\c#\python.exe D:/Deep/PyCharm/Blog/Test01.py
[3 5 7]

Process finished with exit code 0
  
```



# Example

- Tensor



PYTORCH

array

Tensor

reshape

view(reshape)

linspace

linspace

ones, zeros

ones, zeros

random.randn, random.rand

randn, rand

to, cuda

from\_numpy



numpy



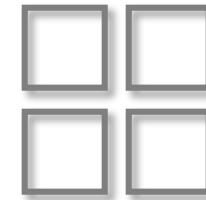
# Example

- Tensor

```
import torch as tc

#Tensor
a = tc.Tensor([2.5, 4])
b = tc.Tensor([[1, 2.5], [2.5, 6]])
```

Tensor a :  
**tensor([ 2.5000, 4.0000])**  
 Size of a : torch.Size([2])  
 Tensor b :  
**tensor([[ 1.0000, 2.5000],  
 [ 2.5000, 6.0000]])**  
 Size of b : torch.Size([2, 2])



```
#reshape
c = b.reshape(1, 4)
d = b.reshape(1, -1)
```

Tensor c :  
**tensor([[ 1.0000, 2.5000, 2.5000, 6.0000]])**  
 Size of c : torch.Size([1, 4])  
 Tensor d :  
**tensor([[ 1.0000, 2.5000, 2.5000, 6.0000]])**  
 Size of d : torch.Size([1, 4])

# Example

- Tensor

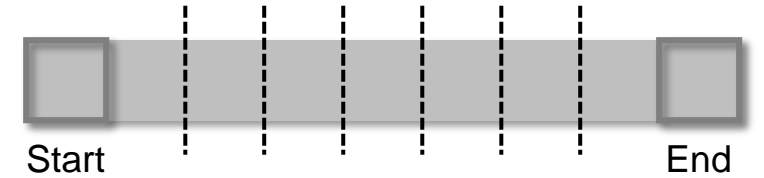
`#linspace`

`x = tc.linspace(-1, 1, 9)`

Tensor x :

`tensor([-1.0000, -0.7500, -0.5000,  
-0.2500, 0.0000, 0.2500, 0.5000,  
0.7500, 1.0000])`

Size of x : `torch.Size([9])`



`#ones, zeros`

`real = tc.ones(2, 2)`

`fake = tc.zeros(2, 2)`

Tensor real :

`tensor([[ 1., 1.],  
[ 1., 1.]])`

Size of real : `torch.Size([2, 2])`

Tensor fake :

`tensor([[ 0., 0.],  
[ 0., 0.]])`

Size of fake : `torch.Size([2, 2])`

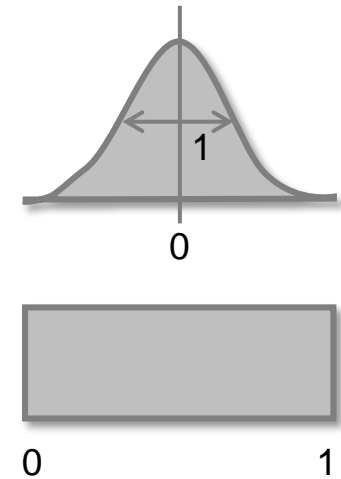


# Example

- Tensor

```
#randn, rand
z1 = tc.randn(2, 3)
z2 = tc.rand(2, 3)
```

```
Tensor z1 :
tensor([[ -0.0750, -1.0555, -0.0706],
        [-0.1946, -0.8593, -0.2238]])
Size of z1 : torch.Size([2, 3])
Tensor z2 :
tensor([[ 0.8204, 0.3505, 0.1034],
        [ 0.5318, 0.9543, 0.8781]])
Size of z2 : torch.Size([2, 3])
```



```
#to, cuda
z_cuda = z1.cuda()
device = tc.device('cuda' if tc.cuda.is_available() else 'cpu')
z_device = z1.to(device)
```

```
Tensor z_cuda :
tensor([[ -0.0750, -1.0555, -0.0706],
        [-0.1946, -0.8593, -0.2238]], device='cuda:0')
Tensor z_device :
tensor([[ -0.0750, -1.0555, -0.0706],
        [-0.1946, -0.8593, -0.2238]], device='cuda:0')
```

# Example

- Tensor

```
#from_numpy, numpy  
import numpy as np
```

```
a = np.array([3.5, 4])  
b = tc.from_numpy(a)  
c = b.numpy()
```



```
array a :  
[3.5 4. ]  
Tensor b :  
tensor([ 3.5000, 4.0000], dtype=torch.float64)  
array c :  
[3.5 4. ]
```

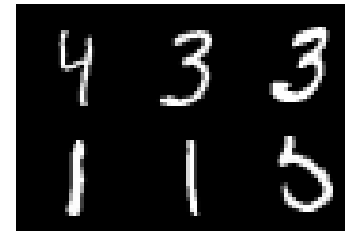
# Example

- Datasets

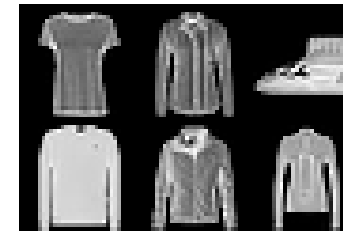
The following datasets are available:

- ✓ MNIST
- ✓ Fashion-MNIST
- EMNIST
- COCO
- ✓ LSUN
- ✓ ImageFolder
- DatasetFolder
- Imagenet-12
- ✓ CIFAR
- STL10
- SVHN
- PhotoTour

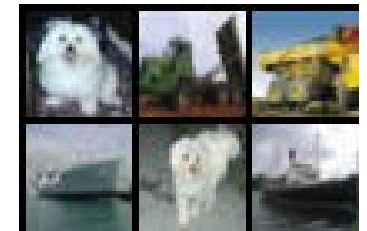
- ✓ : 바로 사용 가능.
- ✓ : 추가적인 과정 필요.
- ✓ : Custom dataset에 활용.



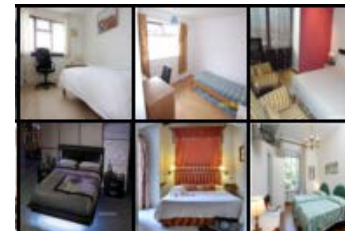
MNIST



Fashion-MNIST



CIFAR



LSUN



CelebA



K-pop(Custom)

# Example

- Datasets – MNIST

```
import torch as tc
import torchvision as tv
import torchvision.transforms as transforms
```

```
trans = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])
datasets = tv.datasets.MNIST(root='./MNIST', train=True, download=True, transform=trans)
dataloader = tc.utils.data.DataLoader(datasets=datasets, batch_size=100, shuffle=True)
```

## trans

- `Compose()` is used when there are multiple transform options. Here, `ToTensor()` and `Normalize(mean, std)` are used.
- `ToTensor()` changes the PIL Image to a tensor. torchvision dataset The default type is PIL Image.
- `Normalize (mean, std)` transforms the range of the image. Here, the value of `[0, 1]` is adjusted to `[-1, 1]`.  $((\text{value}-\text{mean}) / \text{std})$

# Example

- Datasets – MNIST

```
import torch as tc
import torchvision as tv
import torchvision.transforms as transforms
```

```
trans = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])
datasets = tv.datasets.MNIST(root='./MNIST', train=True, download=True, transform=trans)
dataloader = tc.utils.data.DataLoader(datasets=datasets, batch_size=100, shuffle=True)
```

## dataset

- root : This is the path to store (MNIST data). Folders are automatically created with the specified name.
- train : Set the data to be used for the train.
- transform : Transform the data according to the transform option set previously.
- download : Download (MNIST data). (If you downloaded it once, it will not do it again.)



# Example

- Datasets – MNIST

```
import torch as tc
import torchvision as tv
import torchvision.transforms as transforms
```

```
trans = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])
datasets = tv.datasets.MNIST(root='./MNIST', train=True, download=True, transform=trans)
dataloader = tc.utils.data.DataLoader(datasets=datasets, batch_size=100, shuffle=True)
```

## dataloader

- dataset : Set the dataset to load.
- batch\_size : Set the batch size.
- shuffle : Shuffle the data and load it.

# Example

- Datasets – LSUN

<https://github.com/fyu/lsun>

fyu / [lsun](#) Watch 12 Star 144 Fork 66

Code Issues 11 Pull requests 4 Projects 0 Wiki Insights

LSUN Dataset Documentation and Demo Code

22 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file **Clone or download**

fyu Merge branch 'master' of github.com:fyu/lsun\_toolkit Latest commit 7264941 on 4 Jul 2016

README.md	Update README.md	2 years ago
category_indices.txt	add submission details	3 years ago
data.py	support flat export directory	2 years ago
download.py	add downloading category	3 years ago

# Example

- Datasets – LSUN

Run `download.py`    **`download.py -c bedroom`**

{  
 bedroom 0  
 bridge 1  
 church\_outdoor 2  
 classroom 3  
 conference\_room 4  
 dining\_room 5  
 kitchen 6  
 living\_room 7  
 restaurant 8  
 tower 9

If you are using Python 3.0 or later, modify the code from `urllib2.urlopen (url)` to `urlopen (url)`.

```

def list_categories(tag):
    url = 'http://lsun.cs.princeton.edu/htbin/list.cgi?tag=' + tag
    f = urlopen(url)
    return json.loads(f.read())
  
```

# Example

- Datasets – LSUN

```
trans = transforms.Compose([transforms.Resize((64,64)) ,transforms.ToTensor(), transforms.Normalize((0.5, 0.5, 0.5),  
(0.5, 0.5, 0.5))])  
datasets = tv.datasets.LSUN('.', classes=['bedroom_train'], transform=trans)  
dataloader = tc.utils.data.DataLoader(datasets=datasets, batch_size=100, shuffle=True)
```

## trans

- `Resize()` is used to resize the image.

## datasets

- `root` : Root directory for the database files.
- `classes` : One of {'train', 'val', 'test'} or a list of categories to load. e.g. ['bedroom\_train', 'church\_train'].

# Example

- Datasets – ImageFolder(CelebA)

<http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>

## News

2016-07-29 If DropBox are not accessible, please download the dataset using [Google Drive](#) or [Baidu Drive](#).

## Details

**CelebFaces Attributes Dataset (CelebA)** is a large-scale face attributes dataset with more than **200K** celebrity images, each with **40** attribute annotations. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, including

- **10,177** number of **identities**,
- **202,599** number of **face images**, and
- **5 landmark locations**, **40 binary attributes** annotations per image.

The dataset can be employed as the training and test sets for the following computer vision tasks: face attribute recognition, face detection, and landmark (or facial part) localization.

## Sample Images

Eyeglasses



Wearing Hat



# Example

- Datasets – ImageFolder(CelebA)

<http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>



압축을 풀면 jpg 형태로 파일들이 저장된 것을 확인할 수 있음.

# Example

- Datasets – ImageFolder(CelebA)

```
trans = transforms.Compose([transforms.Resize((64,64)) ,transforms.ToTensor(), transforms.Normalize((0.5, 0.5, 0.5),  
(0.5, 0.5, 0.5))])  
datasets = tv.datasets.ImageFolder('./img_align_celeba', trans)  
dataloader = tc.utils.data.DataLoader(datasets=datasets, batch_size=100, shuffle=True)
```

## datasets

- root : Root directory for the database files.
- transform : Transform the data according to the transform option set previously.

# Example

- Neural Nets

## Class base

```
class Model(tc.nn.Module):
    def __init__(self):
        super(Model, self).__init__()
        self.linear1 = tc.nn.Linear(D_in, H),
        self.linear2 = tc.nn.Linear(H, D_out)

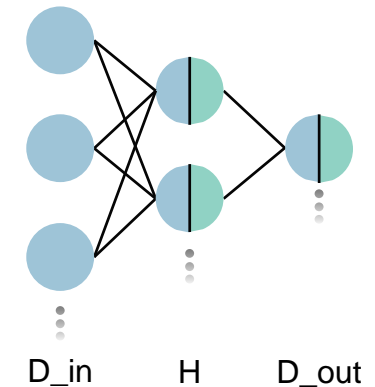
    def forward(self, input):
        x = tc.nn.functional.relu(self.linear1(input))
        x = tc.nn.functional.sigmoid(self.linear2(x))

        return x
```

 Data manipulation

## Sequence base

```
model = tc.nn.Sequential(
    tc.nn.Linear(D_in, H),
    tc.nn.ReLU(),
    tc.nn.Linear(H, D_out),
    tc.nn.Sigmoid()
)
```

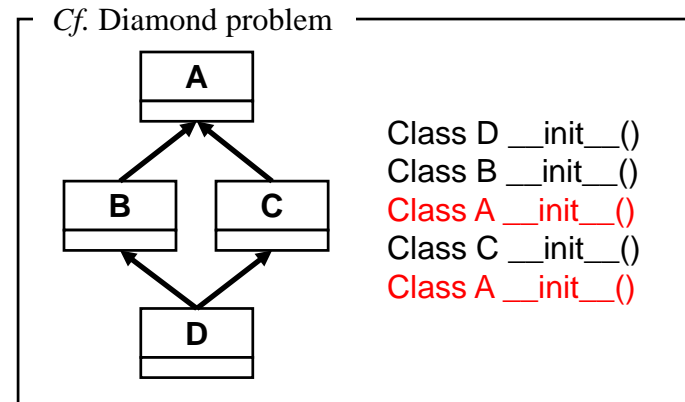




# Example

- Neural Nets

## Class base



```

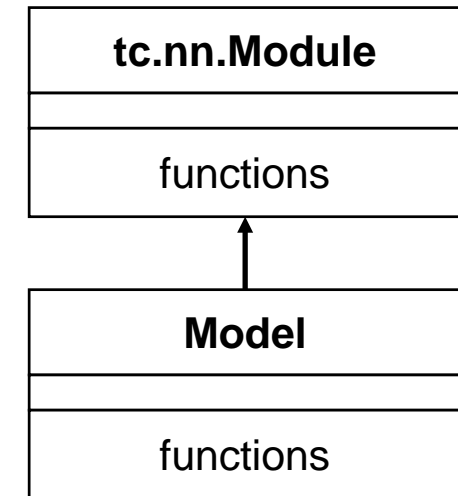
class Model(tc.nn.Module):
    def __init__(self):
        super(Model, self).__init__()
        self.linear1 = tc.nn.Linear(D_in, H),
        self.linear2 = tc.nn.Linear(H, D_out)

    def forward(self, input):
        x = tc.nn.functional.relu(self.linear1(input))
        x = tc.nn.functional.sigmoid(self.linear2(x))

        return x
    
```

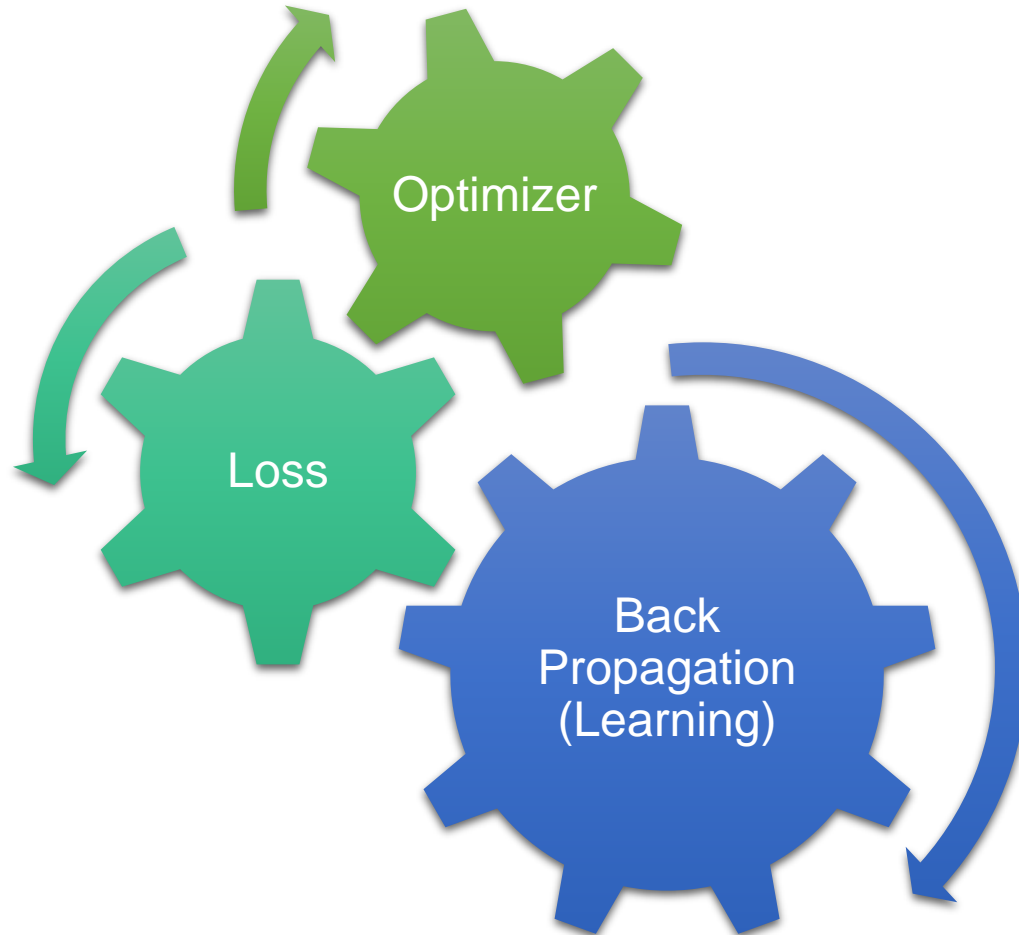
생성자  
부모 클래스 초기화

Propagation



# Example

- Learning



```
loss_func = tc.nn.MSELoss()
opt = tc.optim.Adam(model.parameters(), lr=0.01 )

for ep in range(epoch_sz):
    for step, (images, labels) in enumerate(dataloader):
        opt.zero_grad()
        images = images.to(device)
        labels = labels.to(device)

        output = model(images)

        loss = loss_func(output)
        loss.backward()
        opt.step()
```

# Example

- Learning

*Cf.*

```
trans = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])
datasets = tv.datasets.MNIST(root='./MNIST', train=True, download=True, transform=trans)
dataloader = tc.utils.data.DataLoader(datasets=datasets, batch_size=100, shuffle=True)
```

```
loss_func = tc.nn.MSELoss()
```

```
opt = tc.optim.Adam(model.parameters(), lr=0.01 )
```

```
for ep in range(epoch_sz):
```

```
    for step, (images, labels) in enumerate(dataloader):
```

```
        opt.zero_grad()
```

```
        images = images.to(device)
```

```
        labels = labels.to(device)
```

```
        output = model(images)
```

```
        loss = loss_func(output)
```

```
        loss.backward()
```

```
        opt.step()
```

→ Loss 선언

→ Optimizer 선언 (업데이트 하려는 parameter)

→ 정해진 epoch 만큼 수행

→ 앞서 설정한 dataloader에 따라 image(data)와 label(\_) 불러옴

→ Gradient 초기화.

→ Loss 구함

→ Back propagation(Gradient 구함)

→ 정해진 optimizer에 따라 parameter 업데이트

# Example

- Learning

## Loss

- |                           |                            |
|---------------------------|----------------------------|
| • L1Loss                  | • SmoothL1Loss             |
| • <b>MSELoss</b>          | • SoftMarginLoss           |
| • <b>CrossEntropyLoss</b> | • MultiLabelSoftMarginLoss |
| • NLLLoss                 | • CosineEmbeddingLoss      |
| • PoissonNLLLoss          | • MultiMarginLoss          |
| • KLDivLoss               | • TripleMarginLoss         |
| • <b>BCELoss</b>          |                            |
| • BCEWithLogitsLoss       |                            |
| • MarginRankingLoss       |                            |
| • HingeEmbeddingLoss      |                            |

## Optimizer

- |               |
|---------------|
| • Adadelta    |
| • Adagrad     |
| • <b>Adam</b> |
| • SparseAdam  |
| • Adamax      |
| • ASGD        |
| • LBFGS       |
| • RMSprop     |
| • Rprop       |
| • <b>SGD</b>  |

\* Bold체는 주로 사용하는 함수들.

# Example

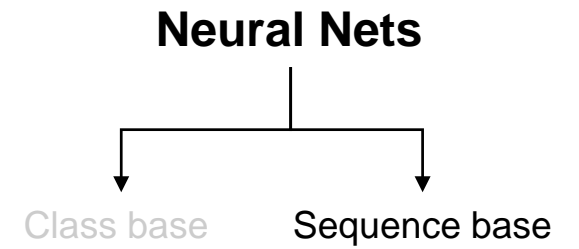
- Applications –Vanilla GAN

*Cf.*

**“What is this? Gum? It's GAN.”**

# Example

- Applications –Vanilla GAN



# Example

- Applications –Vanilla GAN

```
loss_func = tc.nn.BCELoss()  
d_opt = tc.optim.Adam(D.parameters(), lr=lr)  
g_opt = tc.optim.Adam(G.parameters(), lr=lr)
```

# Example

- Applications –Vanilla GAN

```

for ep in range(nEpoch):
    for step, (images, ) in enumerate(dataloader):
        images = images.reshape(batch_sz, -1).to(device)
        z = tc.randn(batch_sz, noise_sz).to(device)

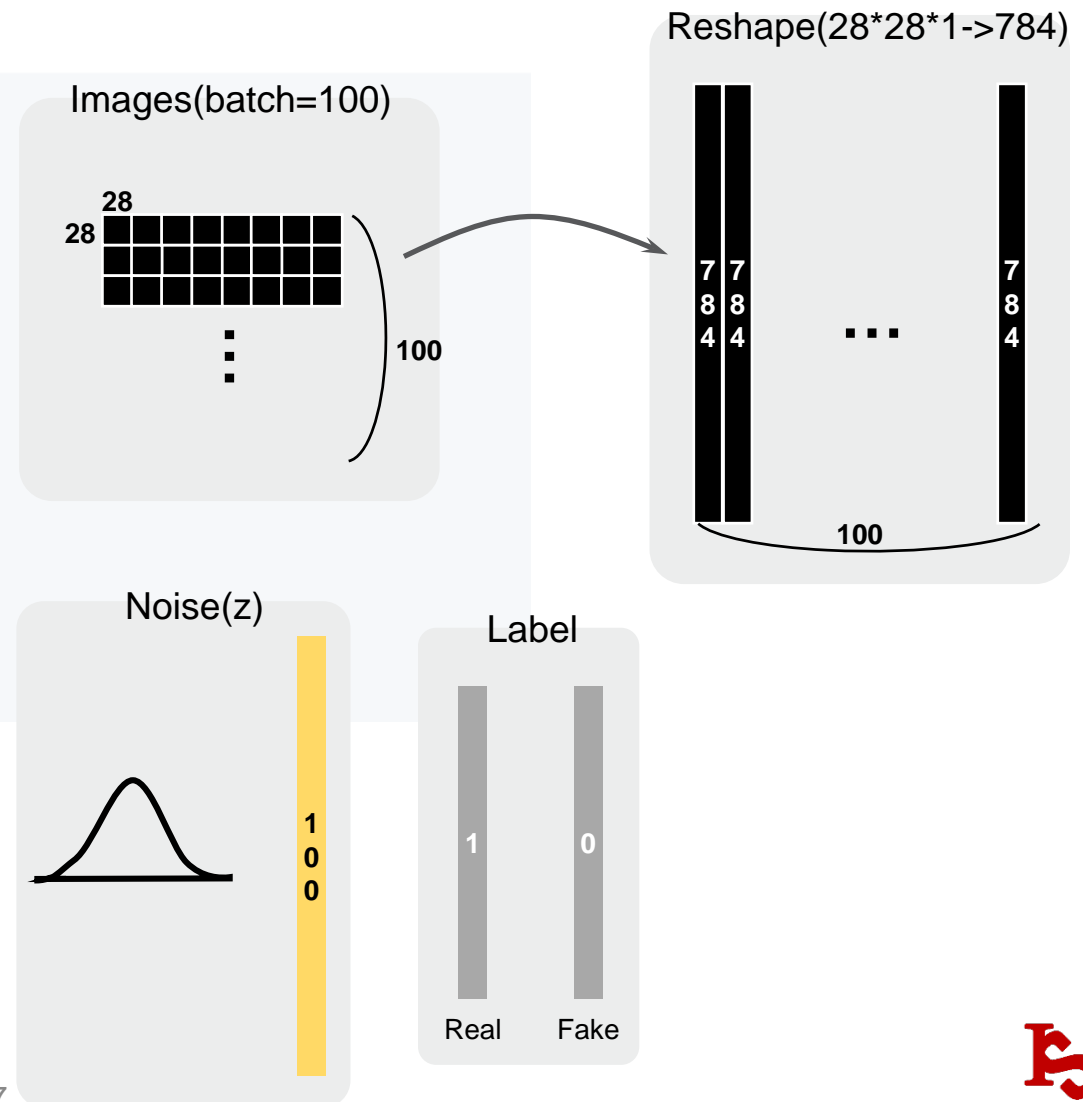
        real_label = tc.ones(batch_sz, 1).to(device)
        fake_label = tc.zeros(batch_sz, 1).to(device)

        loss_real = loss_func(D(images), real_label)
        loss_fake = loss_func(D(G(z)), fake_label)

        d_loss = loss_real + loss_fake

        d_opt.zero_grad()
        d_loss.backward()
        d_opt.step()

```





# Example

- Applications –Vanilla GAN

```

for ep in range(nEpoch):
    for step, (images, _) in enumerate(dataloader):
        images = images.reshape(batch_sz, -1).to(device)
        z = tc.randn(batch_sz, noise_sz).to(device)

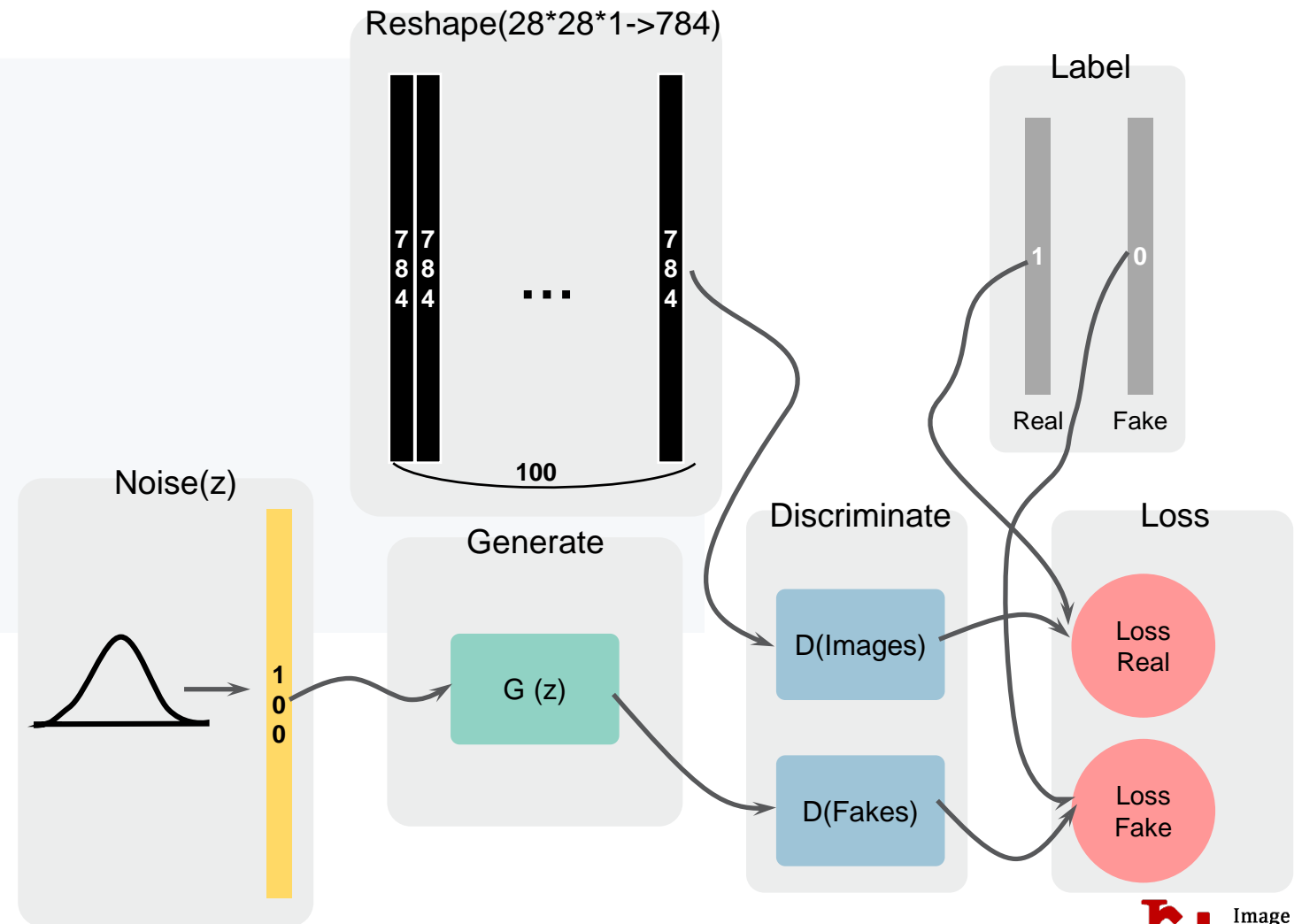
        real_label = tc.ones(batch_sz, 1).to(device)
        fake_label = tc.zeros(batch_sz, 1).to(device)

        loss_real = loss_func(D(images), real_label)
        loss_fake = loss_func(D(G(z)), fake_label)

        d_loss = loss_real + loss_fake

        d_opt.zero_grad()
        d_loss.backward()
        d_opt.step()

```



# Example

- Applications –Vanilla GAN

```

for ep in range(nEpoch):
    for step, (images, _) in enumerate(dataloader):
        images = images.reshape(batch_sz, -1).to(device)
        z = tc.randn(batch_sz, noise_sz).to(device)

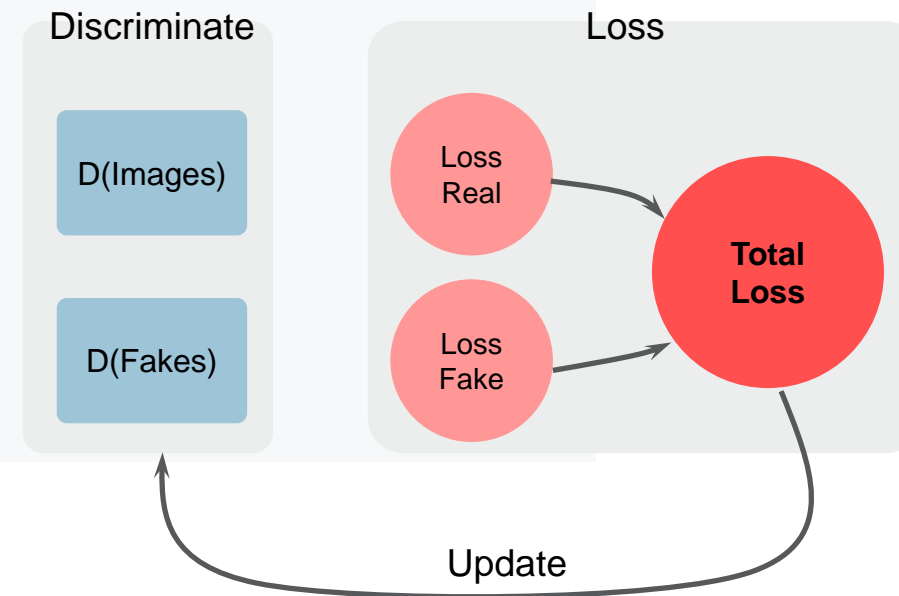
        real_label = tc.ones(batch_sz, 1).to(device)
        fake_label = tc.zeros(batch_sz, 1).to(device)

        loss_real = loss_func(D(images), real_label)
        loss_fake = loss_func(D(G(z)), fake_label)

        d_loss = loss_real + loss_fake

        d_opt.zero_grad()
        d_loss.backward()
        d_opt.step()

```



# Example

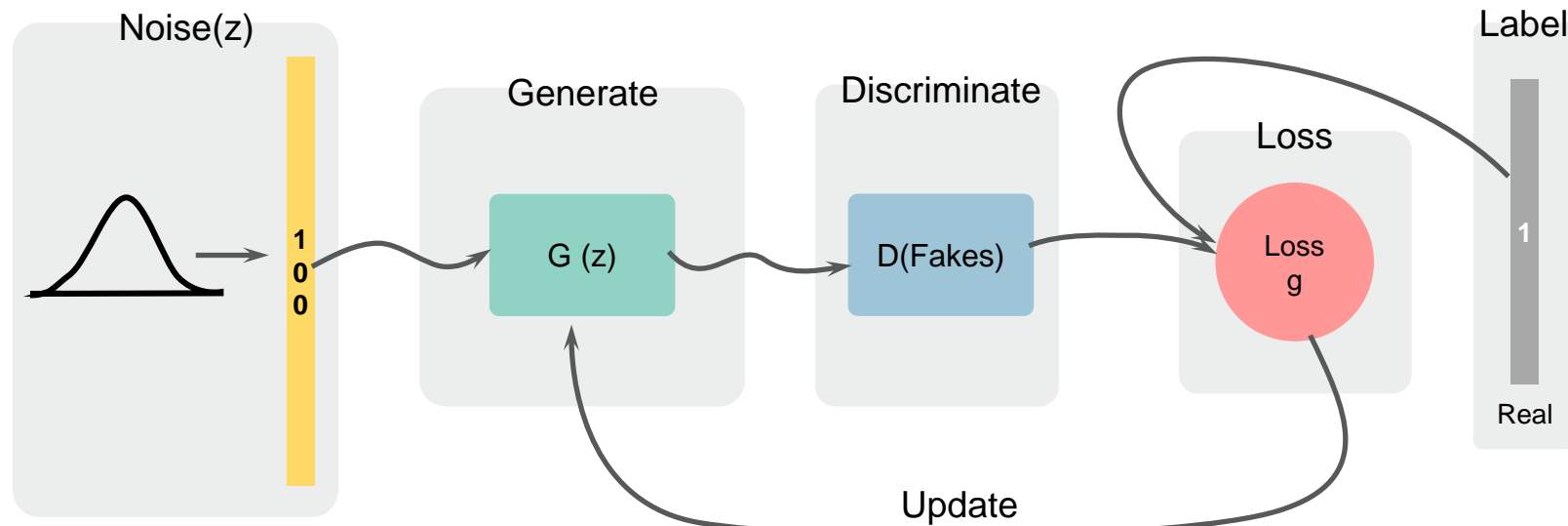
- Applications –Vanilla GAN

```

fake_images = G(z)
g_loss = loss_func(D(fake_images), real_label)

g_opt.zero_grad()
g_loss.backward()
g_opt.step()

```



# Future Work

## Paper Review



- Vanilla GAN
- DCGAN
- InfoGAN
- Unrolled GAN
- Wasserstein GAN
- LS GAN
- BEGAN
- Pix2Pix
- Cycle GAN

## Proposed Model



- SpyGAN

## Tools



- Document
- Programming
- PyTorch
- Python executable & UI

## Mathematical Study



- Linear algebra
- Probability and statistics
- Information theory

## Others



- Level Processor
- Ice Propagation

Maybe next seminar?

Q

&

A

*Thank you for your attention*