

# The Software: PyTorch, fastai, and Jupyter

- 많은 딥러닝 라이브러리들 중 PyTorch를 선택
  - simplicity와 speed 양쪽을 잡은 라이브러리
- fast.ai는 PyTorch 기반 higher-level functionality를 제공(이 책은 fastai v2 사용)
- 어떤 라이브러리를 사용하는가 자체는 크게 중요하지 않음
  - 하나에서 다른 하나로 넘어가는 것은 며칠이면 가능
  - 고수준 개념 설명에는 fastai를, 저수준 개념 설명에는 pytorch나 순수 python 사용 예정
  - (어차피 몇년 있으면 또 새로운 게 나올 것... ex. web programming)
- fastai의 사용법과 내부 구현까지를 모두 이해하는 것이 목표
- 실습 환경으로는 Jupyter Notebook을 추천

## Your First Model

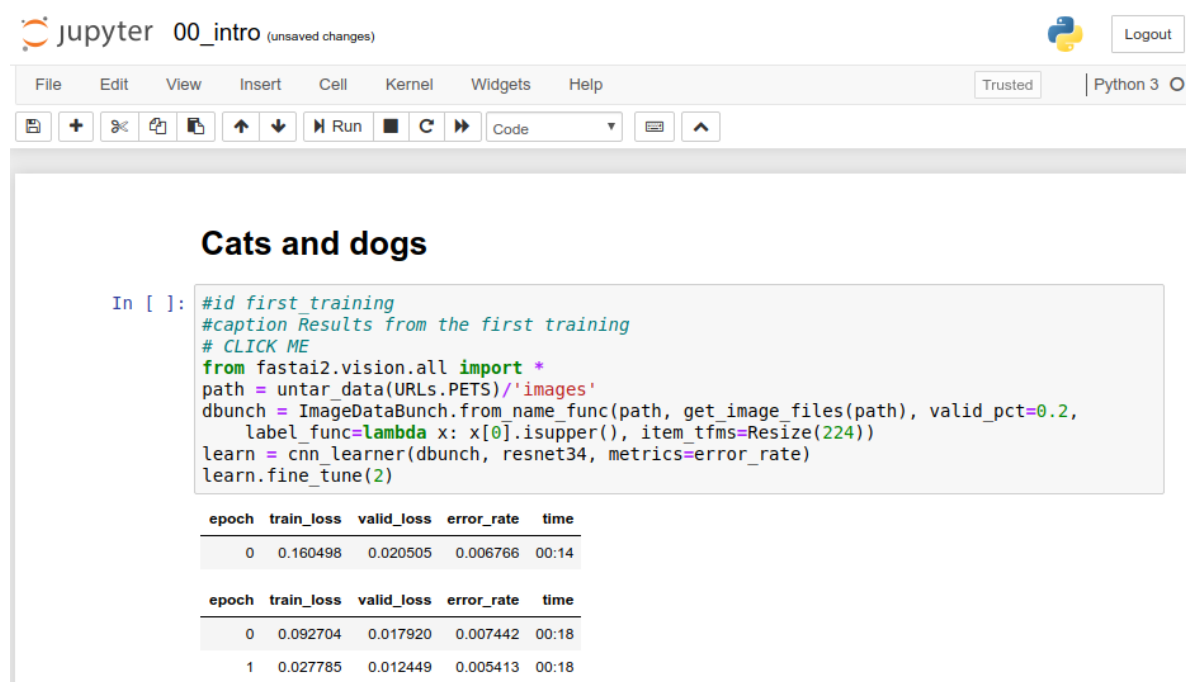
고양이vs강아지를 구분하는 모델을 구현하면서 top-down 방식으로 설명 진행

## Getting a GPU Deep Learning Server

- 실습에는 NVIDIA GPU 필요
- 개인 차원에서 실물 GPU를 구매하는 것보다는 클라우드 서비스를 이용하는 것을 추천
  - 리눅스 서버를 세팅하고 각종 라이브러리를 수동으로 설치하는 것은 비효율적

## Running Your First Notebook

Jupyter notebook을 사용해 dataset을 받고 모델을 학습하는 예제



The screenshot shows a Jupyter Notebook interface with the title "00\_intro (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a status bar indicating "Trusted" and "Python 3".

The active cell is a code cell with the following content:

```
In [ ]: #id first training
#caption Results from the first training
# CLICK ME
from fastai2.vision.all import *
path = untar_data(URLs.PETS)/'images'
dbunch = ImageDataBunch.from_name_func(path, get_image_files(path), valid_pct=0.2,
    label_func=lambda x: x[0].isupper(), item_tfms=Resize(224))
learn = cnn_learner(dbunch, resnet34, metrics=error_rate)
learn.fine_tune(2)
```

The output of the code cell is a table showing training results:

epoch	train_loss	valid_loss	error_rate	time
0	0.160498	0.020505	0.006766	00:14

epoch	train_loss	valid_loss	error_rate	time
0	0.092704	0.017920	0.007442	00:18
1	0.027785	0.012449	0.005413	00:18

- Code Cell : 파이썬 코드 작성 가능. 각 셀의 출력 결과는 셀의 아래쪽에 표시됨
- Text Cell : 마크다운 문법을 이용해 텍스트 편집 가능

<코드>

```

from fastai2.vision.all import *
path = untar_data(URLs.PETS)/'images'

def is_cat(x): return x[0].isupper()

dls = ImageDataLoaders.from_name_func(
    path, get_image_files(path), valid_pct=0.2, seed=42,
    label_func=is_cat, item_tfms=Resize(224))

learn = cnn_learner(dls, resnet34, metrics=error_rate)
learn.fine_tune(1)

```

위 코드는 다음 동작을 수행한다.

1. Oxford-IIIT Pet Dataset에서 7,349장의 고양이/강아지 사진 데이터를 다운로드 후 압축해제
2. *pretrained model*을 다운로드(`resnet34`)
3. 2.에서 받은 pretrained model에 대해 fine tuning 진행

학습이 완료된 모델에 대해서는 다음과 같이 prediction을 진행할 수 있다.

```

# 혹은 uploader = widgets.FileUpload()
uploader = SimpleNamespace(data = ['images/chapter1_cat_example.jpg'])

img = PILImage.create(uploader.data[0])
is_cat,_,probs = learn.predict(img)
print(f"Is this a cat?: {is_cat}.")
# >> Is this a cat?: True.
print(f"Probability it's a cat: {probs[1].item():.6f}")
# >> Probability it's a cat: 0.999986

```