

Lab2: Build an ANN

11210IPT 553000

Deep Learning in Biomedical Optical Imaging

2023/09/25

Outlines

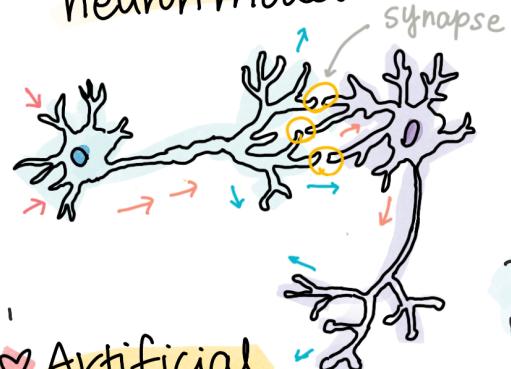
- ▶ Deep Learning Frameworks
- ▶ Kaggle & Dataset

Introduction to

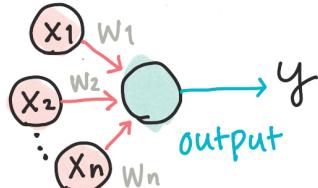
Neural Networks

@girlie_mac
@Shwars

♡ Biological neuron model



♡ Artificial

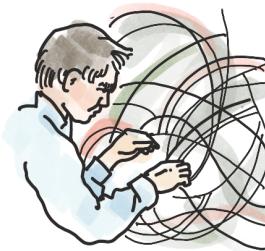


$$y = f\left(\sum_{i=1}^n x_i w_i\right)$$

Perception

= single-layer neural network

1957 Frank Rosenblatt



Mark-I
Perception
computer

- 400 inputs
- 1 binary output

Recognize
Primitive
shapes

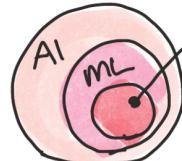
Frameworks

TensorFlow



Low-level
APIs

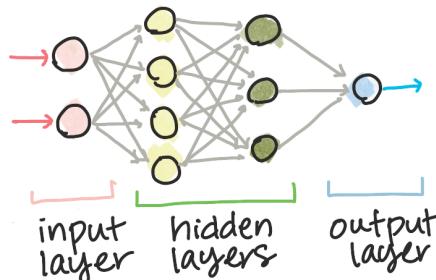
- build computational graph



Neural Networks
↳ Deep Learning

Back propagation
method to train

↔ multi-layer
perception



Keras

High-level APIs

source

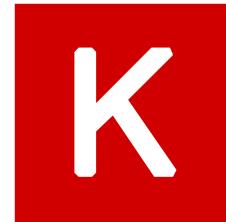
Deep Learning Frameworks



TensorFlow



 PyTorch



 PyTorch
Lightning



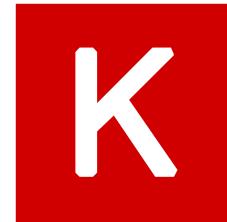
Deep Learning Frameworks

| | Keras  | TensorFlow  | PyTorch  |
|----------------------------------|--|---|--|
| Level of API | high-level API ¹ | Both high & low level APIs | Lower-level API ² |
| Speed | Slow | High | High |
| Architecture | Simple, more readable and concise | Not very easy to use | Complex ³ |
| Debugging | No need to debug | Difficult to debugging | Good debugging capabilities |
| Dataset Compatibility | Slow & Small | Fast speed & large | Fast speed & large datasets |
| Popularity Rank | 1 | 2 | 3 |
| Uniqueness | Multiple back-end support | Object Detection Functionality | Flexibility & Short Training Duration |
| Created By | Not a library on its own | Created by Google | Created by Facebook ⁴ |
| Ease of use | User-friendly | Incomprehensive API | Integrated with Python language |
| Computational graphs used | Static graphs | Static graphs | Dynamic computation graphs ⁵ |

TensorFlow vs Keras: A Comparison: <https://towardsdatascience.com/tensorflow-vs-keras-d51f2d68fdfe>
ML03: PyTorch vs. TensorFlow: <https://medium.com/analytics-vidhya/ml03-9de2f0dbd62d>

Deep Learning Frameworks

```
model = keras.models.Sequential()  
model.add(keras.layers.Dense(1, input_shape=(2,), activation='sigmoid'))  
model.summary()
```



```
model.compile(optimizer=keras.optimizers.SGD(learning_rate=0.2), loss='binary_crossentropy', metrics=[ 'acc' ])
```

Reference

1. <https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/3-NeuralNetworks/05-Frameworks/IntroPyTorch.ipynb>
2. <https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/3-NeuralNetworks/05-Frameworks/IntroKerasTF.ipynb>
3. <https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/3-NeuralNetworks/05-Frameworks/IntroKeras.ipynb>

Deep Learning Frameworks



```
W = tf.Variable(tf.random.normal(shape=(2,1)), dtype=tf.float32)
b = tf.Variable(tf.zeros(shape=(1,), dtype=tf.float32))

learning_rate = 0.1

@tf.function
def train_on_batch(x, y):
    with tf.GradientTape() as tape:
        z = tf.matmul(x, W) + b
        loss = tf.reduce_mean(tf.nn.sigmoid_cross_entropy_with_logits(labels=y, logits=z))
    dloss_dw, dloss_db = tape.gradient(loss, [W, b])
    W.assign_sub(learning_rate * dloss_dw)
    b.assign_sub(learning_rate * dloss_db)
    return loss
```



TensorFlow

Reference

1. <https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/3-NeuralNetworks/05-Frameworks/IntroPyTorch.ipynb>
2. <https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/3-NeuralNetworks/05-Frameworks/IntroKerasTF.ipynb>
3. <https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/3-NeuralNetworks/05-Frameworks/IntroKeras.ipynb>

Deep Learning Frameworks

```
class Network():
    def __init__(self):
        self.W = torch.randn(size=(2,1), requires_grad=True)
        self.b = torch.zeros(size=(1,), requires_grad=True)

    def forward(self,x):
        return torch.matmul(x, self.W)+self.b

    def zero_grad(self):
        self.W.data.zero_()
        self.b.data.zero_()

    def update(self,lr=0.1):
        self.W.data.sub_(lr*self.W.grad)
        self.b.data.sub_(lr*self.b)

net = Network()
```

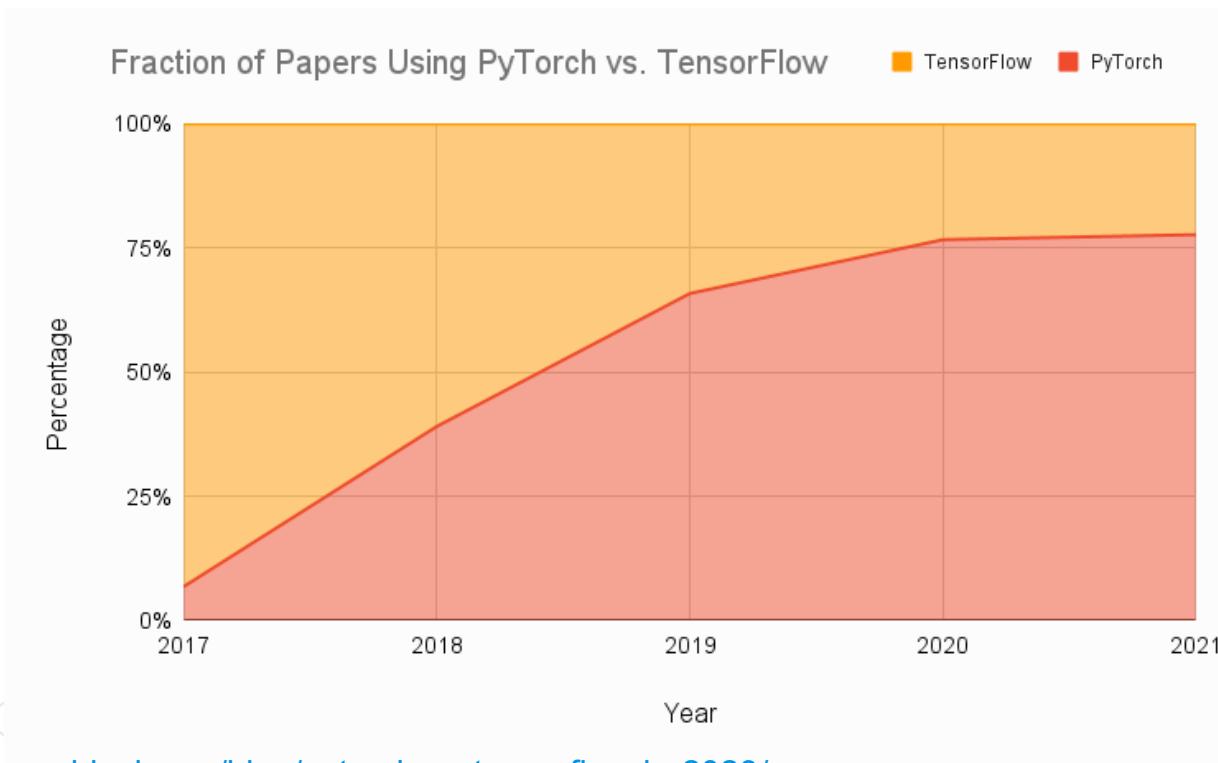


```
def train_on_batch(net, x, y):
    z = net.forward(x).flatten()
    loss = torch.nn.functional.binary_cross_entropy_with_logits(input=z, target=y)
    net.zero_grad()
    loss.backward()
    net.update()
    return loss
```

Reference

1. <https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/3-NeuralNetworks/05-Frameworks/IntroPyTorch.ipynb>
2. <https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/3-NeuralNetworks/05-Frameworks/IntroKerasTF.ipynb>
3. <https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/3-NeuralNetworks/05-Frameworks/IntroKeras.ipynb>

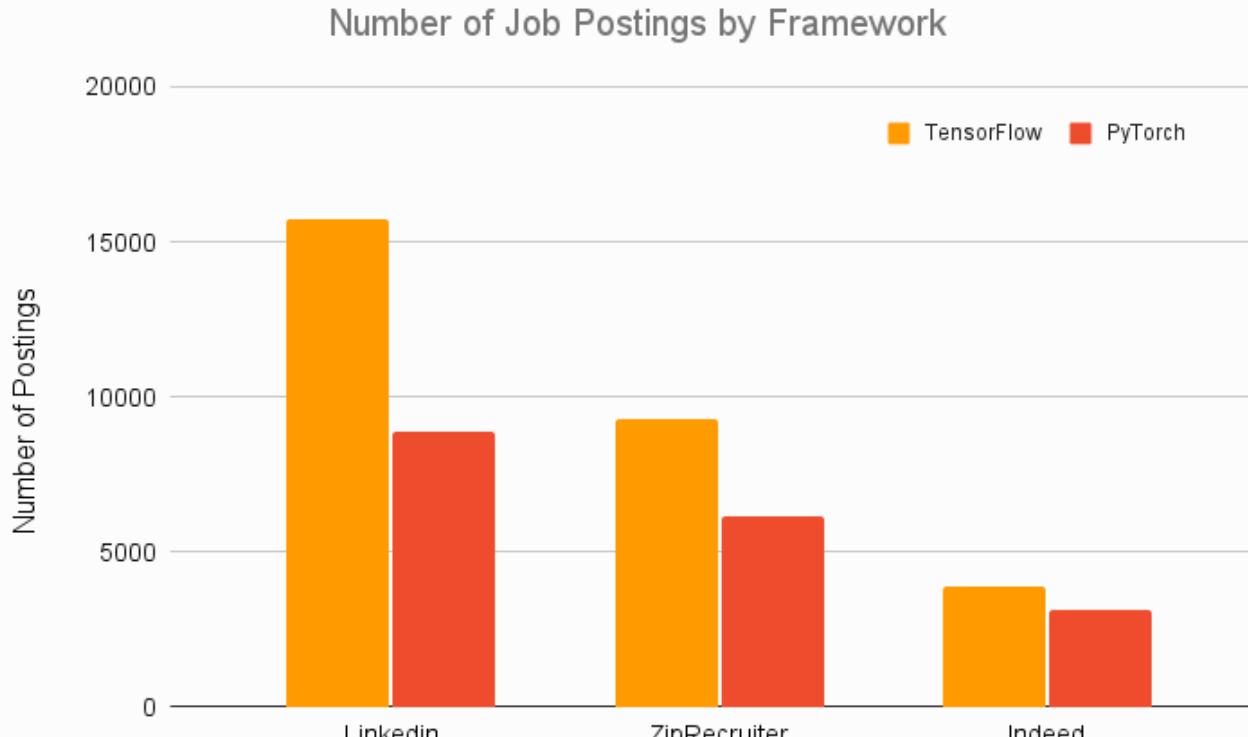
Frameworks - Research Papers



<https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2023/>

<https://thegradient.pub/state-of-ml-frameworks-2019-pytorch-dominates-research-tensorflow-dominates-industry/> 9

Frameworks - Jobs



<https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/>

Kaggle

<https://www.kaggle.com/>

The screenshot shows the Kaggle website's 'Competitions' page. The left sidebar features a navigation menu with 'Competitions' selected. The main content area has a search bar and a large illustration of a person wearing headphones and holding a large gold coin labeled '\$1'. Below the illustration, there are five categories: 'All Competitions', 'Featured', 'Getting Started', 'Research', and 'Community'. The 'Active Competitions' section displays three competition thumbnails. The first thumbnail shows people working at a table. The second thumbnail shows a Google logo and a city skyline. The third thumbnail features a horse and the text 'BIG DATA Derby'.

kaggle

Create

Home

Competitions

Datasets

Code

Discussions

Learn

More

Your Work

RECENTLY VIEWED

Heart_Disease_ML_Cl...

Heart Disease Dataset

RECENTLY EDITED

Heart_Disease_ML_Cl...

Search

Search competitions

Filters

All Competitions

Featured

Getting Started

Research

Community

Hotness

Active Competitions

BIG DATA Derby

Chest X-Ray Image Dataset

- ▶ This is an open dataset from [Kaggle](#).
- ▶ It has 3 folders (train, test, val) and 2 classes (pneumonia/normal) for each category.
- ▶ The original dataset is an imbalanced dataset, the ratio of pneumonia and normal is roughly 2.5:1.
- ▶ For the lab, we took 1000 images for each category and resized all images into 256×256.
- ▶ The pixel values are range by [0, 255].

CODING TIME!!

