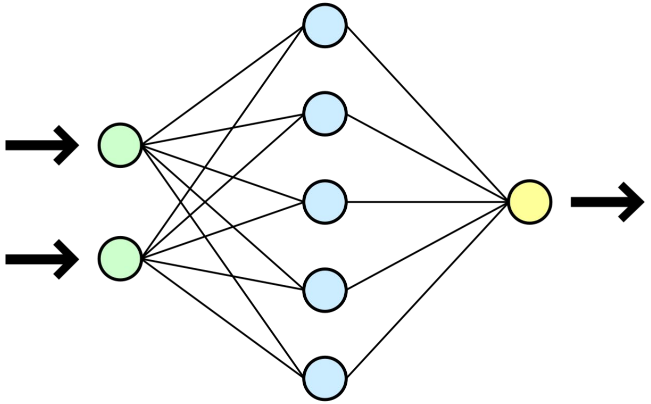


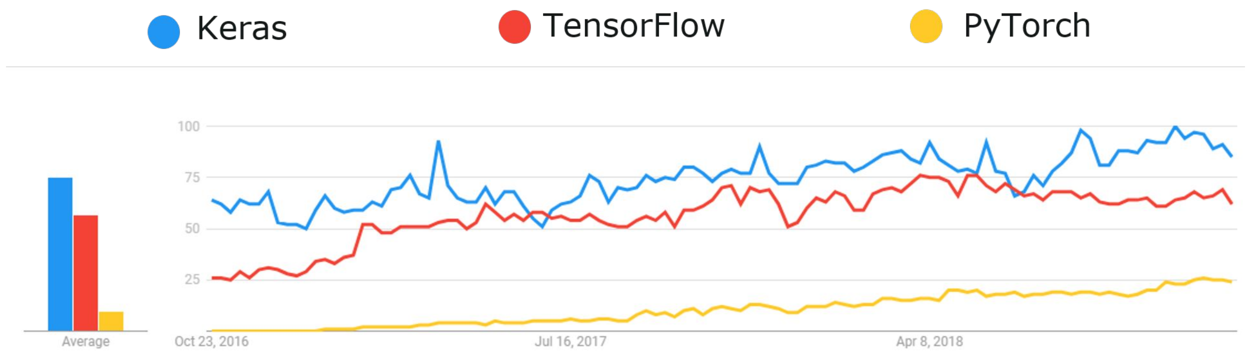
THE CODING LIBRARY

Neural Networks

Keras vs PyTorch - Best Deep Learning Framework

Deep Learning Frameworks

- TensorFlow
- Keras
- PyTorch
- Sonnet
- Caffe
- MXNet
- Gluon
- ONNX
- Chainer
- A lot more



THE CODING LIBRARY

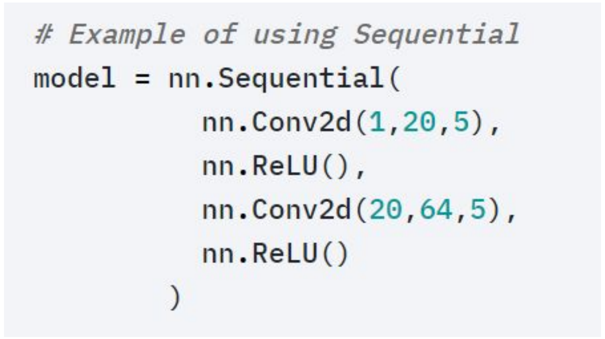
TensorFlow / Keras

- Latest versions of TensorFlow are now closely integrated with Keras
- Keras is a high level simple API that uses TensorFlow as a backend
- TensorFlow is created by the Google Brain Team
- Keras is beginner friendly
- Support for distributed training on GPUs and TPUs
- It’s flexible when creating and deploying models
- AutoKeras - Automatically finding top performing models for your data and application
- Pre-trained models are available
- TensorFlow API available for Python, C++, Javascript, Java and Go

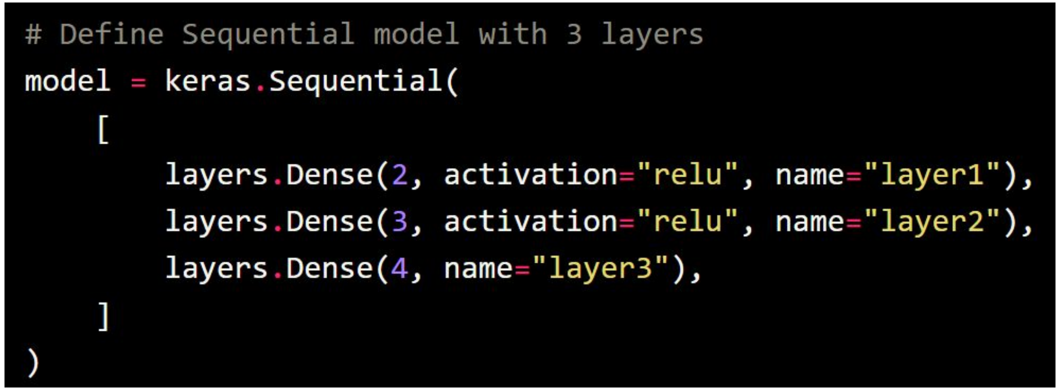
PyTorch

- Created by Facebook AI Research
- Similar to Keras but has a more complex API
- Modern software products - Tesla Autopilot and FSD
- Support for distributed training on GPUs and TPUs
- It’s flexible when creating your own neural networks for research
- Pytorch has a large ecosystem of additional frameworks built on top of it
- Auto-Pytorch - Automatically finding top performing models for your data and application
- Pre-trained models are available
- PyTorch API available for Python, C++ and Java

Keras vs PyTorch - Create a Neural Network



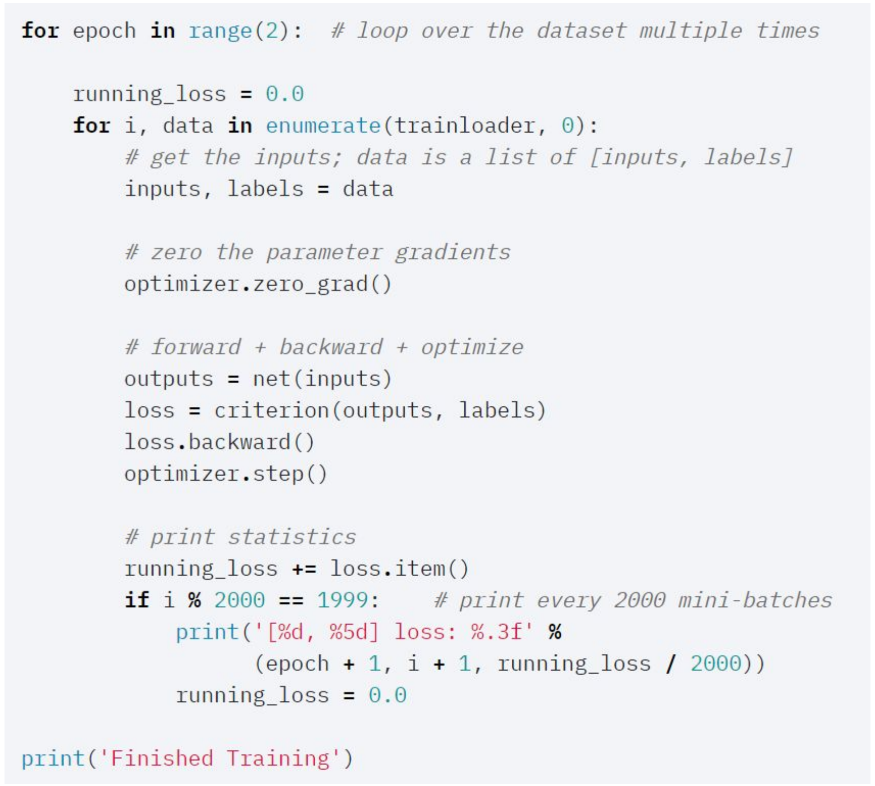
```python
# Example of using Sequential
model = nn.Sequential(
          nn.Conv2d(1,20,5),
          nn.ReLU(),
          nn.Conv2d(20,64,5),
          nn.ReLU()
        )
```



```python
# Define Sequential model with 3 layers
model = keras.Sequential(
    [
        layers.Dense(2, activation="relu", name="layer1"),
        layers.Dense(3, activation="relu", name="layer2"),
        layers.Dense(4, name="layer3"),
    ]
)
```

PyTorch - Training a Neural Network

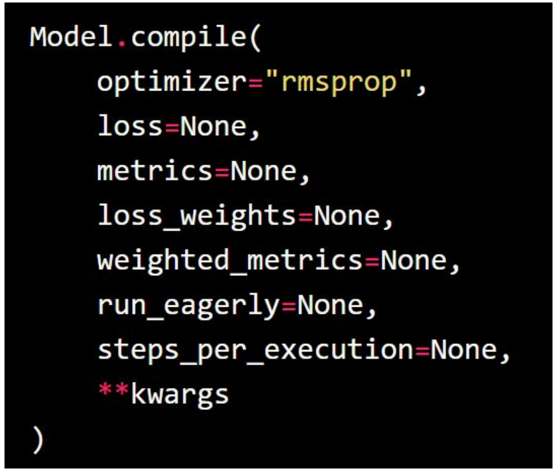
THE CODING LIBRARY



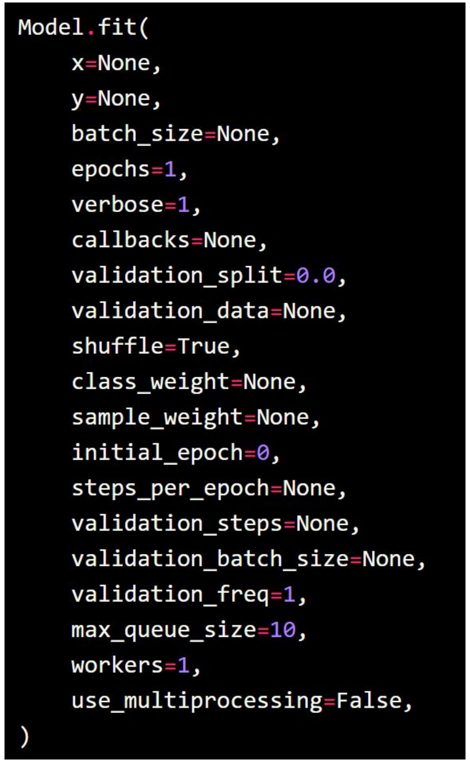
```python
for epoch in range(2):  # loop over the dataset multiple times

    running_loss = 0.0
    for i, data in enumerate(trainloader, 0):
        # get the inputs; data is a list of [inputs, labels]
        inputs, labels = data

        # zero the parameter gradients
        optimizer.zero_grad()

        # forward + backward + optimize
        outputs = net(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

        # print statistics
        running_loss += loss.item()
        if i % 2000 == 1999:    # print every 2000 mini-batches
            print('[%d, %5d] loss: %.3f' %
                  (epoch + 1, i + 1, running_loss / 2000))
            running_loss = 0.0

print('Finished Training')
```

THE CODING LIBRARY

Keras - Train a Neural Network



```
Model.compile(
    optimizer="rmsprop",
    loss=None,
    metrics=None,
    loss_weights=None,
    weighted_metrics=None,
    run_eagerly=None,
    steps_per_execution=None,
    **kwargs
)
```



```
Model.fit(
    x=None,
    y=None,
    batch_size=None,
    epochs=1,
    verbose=1,
    callbacks=None,
    validation_split=0.0,
    validation_data=None,
    shuffle=True,
    class_weight=None,
    sample_weight=None,
    initial_epoch=0,
    steps_per_epoch=None,
    validation_steps=None,
    validation_batch_size=None,
    validation_freq=1,
    max_queue_size=10,
    workers=1,
    use_multiprocessing=False,
)
```

Keras vs PyTorch

- When should one framework be used over the other?

Summary

- Keras/TensorFlow and PyTorch are overall the best and most popular frameworks to use for Deep Learning
- Keras is good to start with if you are a beginner
- PyTorch is good if you are more experienced, you are creating more complex models or doing research