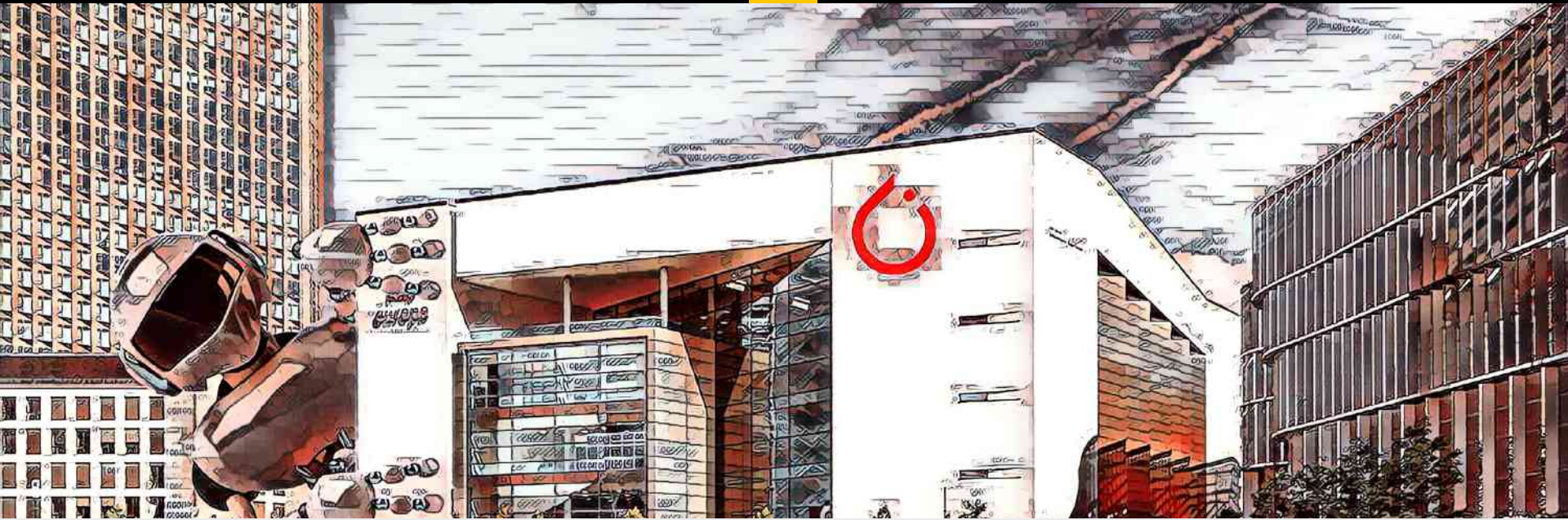
KÖNIGSWEG



IT-Tage 2019

Neuronale Netze mit PyTorch

ALEXANDER CS HENDORF

PyTorch

KÖNIGSWEG

Alexander C. S. Hendorf

ah@koenigsweg.com

@hendorf

—Partner & Principal Consultant Information Technology Consulting on AI & Data Science

—Python Software Foundation Fellow, Python Softwareverband chair, Program Chair of EuroSciPy, PyConDE & PyData Berlin, MongoDB Master, Emeritus EuroPython

—Speaker Europe & USA MongoDB World New York / San José, PyCon Italy, CEBIT Developer World, BI Forum, IT-Tage FFM, PyData London, Berlin, PyParis,... here!

Neuronal Networks with PyTorch

Topics today

- Background of PyTorch
- Tensors
- Computational graphs
- Building an neural net in PyTorch
- Debugging
- Serialization

Origins of PyTorch

Background of PyTorch

- 2017
- Open Source
- Facebook AI & huge community
- Build on Torch, made it popular with Python

Strongsuits

KÖNIGSWEG

Strongsuits of PyTorch

- Easy to learn
- Intuitive
- Integrated with NumPy and SciKit-Learn
- Easy to debug
- Dynamic graphs
- Well documented
- Keeping release schedule

Let’s go

GET STARTED

Select preferences and run the command to install PyTorch locally, or get started quickly with one of the supported cloud platforms.

Start Locally | Start via Cloud Partners | Previous PyTorch Versions | Mobile

Shortcuts

- Prerequisites
  - Supported Linux Distributions
  - Python
  - Package Manager
- Installation
  - Anaconda
  - pip
- Verification
- Building from source
  - Prerequisites

START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.3 builds that are generated nightly. Please ensure that you have met the prerequisites below (e.g., numpy), depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also install previous versions of PyTorch. Note that LibTorch is only available for C++.

PyTorch Build	Stable (1.3)	Preview (Nightly)		
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	9.2	10.1	None	
Run this Command:	conda install pytorch torchvision cudatoolkit=10.1 -c pytorch			

https://pytorch.org/

Tensor Basics

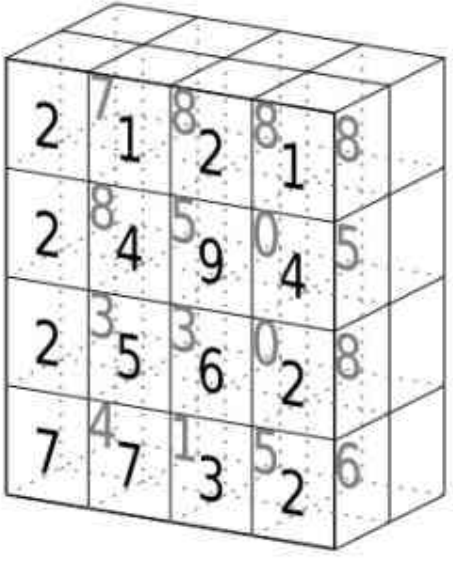
KÖNIGSWEG

't'
'e'
'n'
's'
'o'
'r'

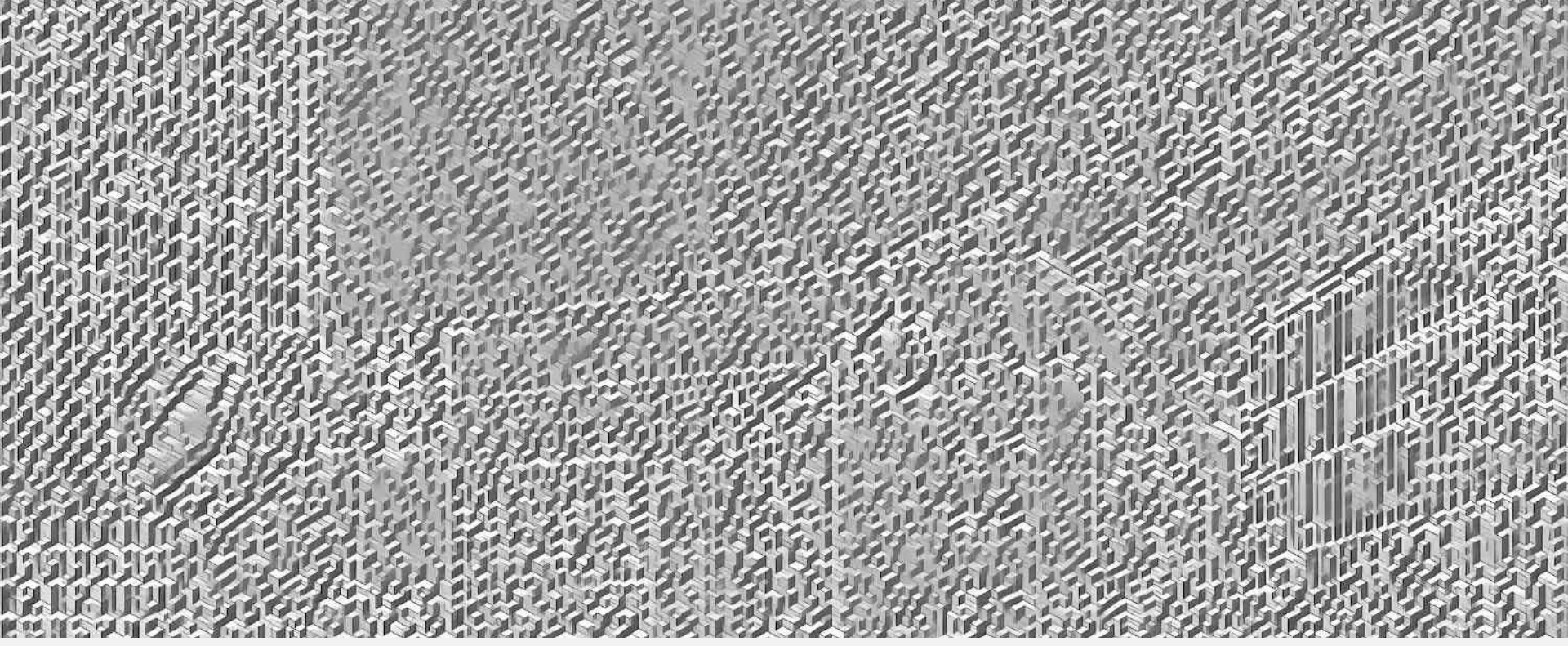
tensor of dimensions [6]
(vector of dimension 6)

3	1	4	1
5	9	2	6
5	3	5	8
9	7	9	3
2	3	8	4
6	2	6	4

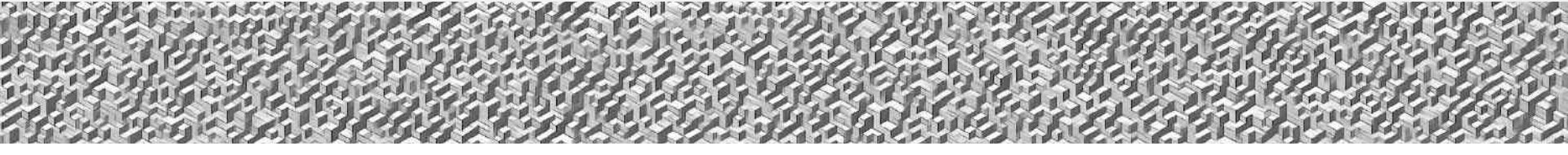
tensor of dimensions [6,4]
(matrix 6 by 4)



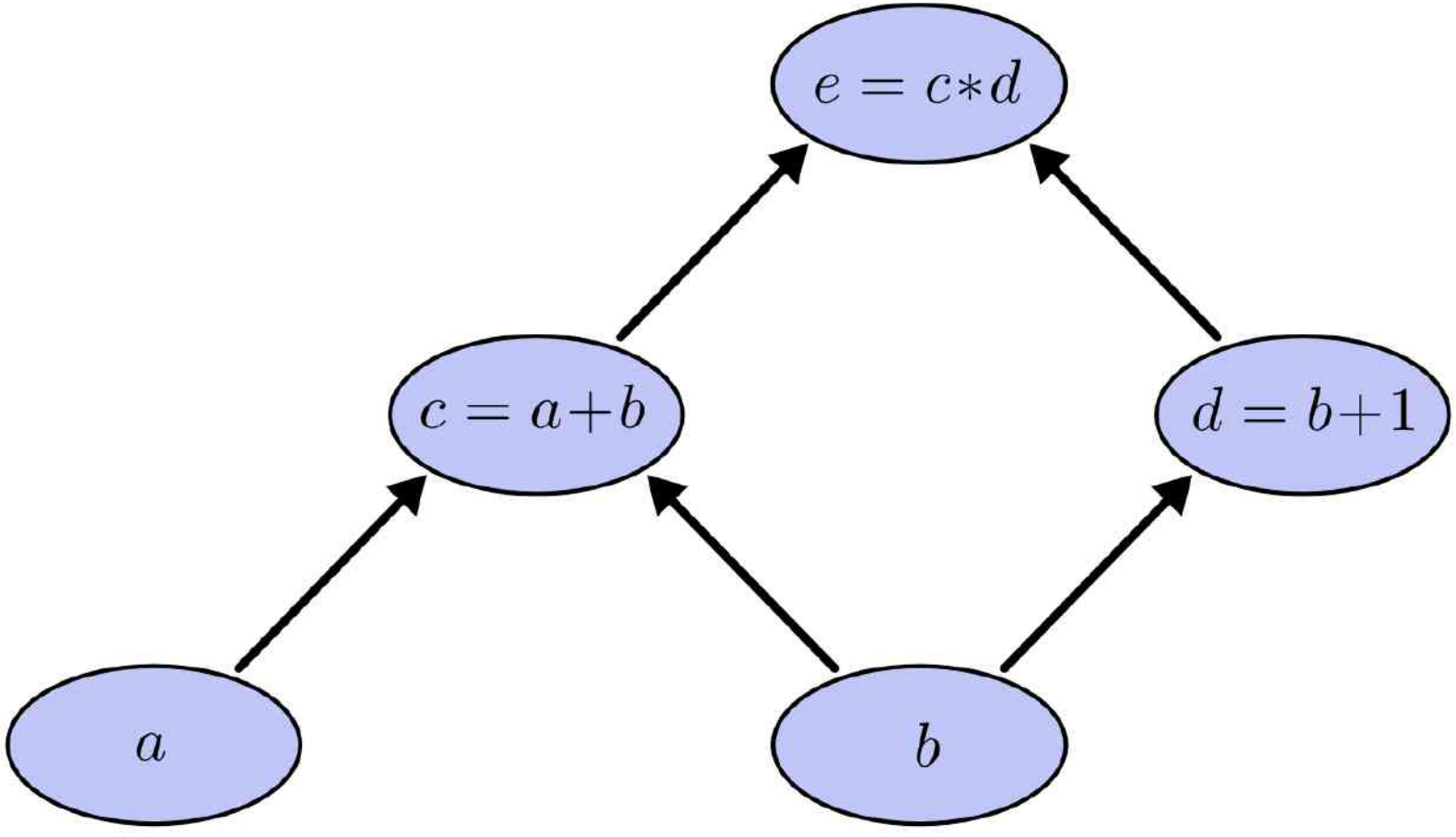
tensor of dimensions [4,4,2]



Computational Graphs



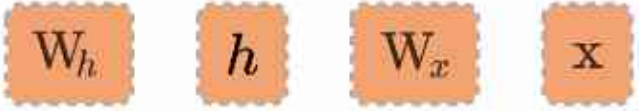
KÖNIGSWEG



https://colah.github.io/

KÖNIGSWEG

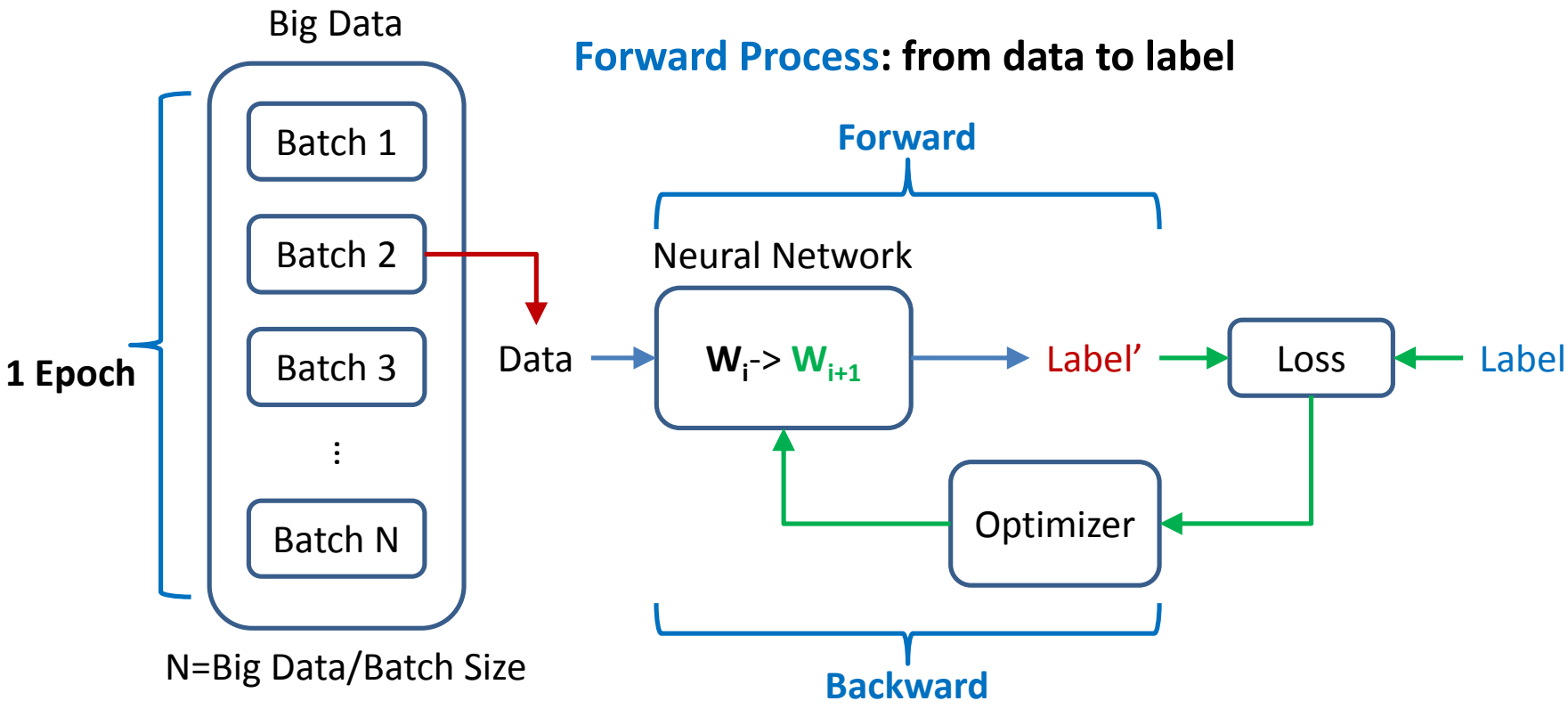
A graph is created on the fly



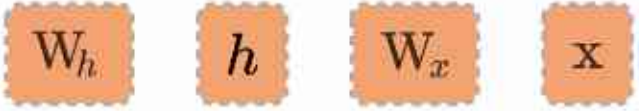

```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))
```

https://pytorch.org


Big Data
Batch 1
Batch 2
Batch 3
Batch N
1 Epoch
N=Big Data/Batch Size
Data
Forward Process: from data to label
Forward
Neural Network
Wi-> Wi+1
Label'
Loss
Label
Optimizer
Backward
Backward Process: update the parameters

A graph is created on the fly




```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))
```

https://pytorch.org

```python
import torch
import torch.nn as nn
import torch.nn.functional as F


class Net(nn.Module):

    def __init__(self):
        super(Net, self).__init__()
        # 1 input image channel, 6 output channels, 3x3 square convolution
        # kernel
        self.conv1 = nn.Conv2d(1, 6, 3)
        self.conv2 = nn.Conv2d(6, 16, 3)
        # an affine operation: y = Wx + b
        self.fc1 = nn.Linear(16 * 6 * 6, 120)  # 6*6 from image dimension
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        # Max pooling over a (2, 2) window
        x = F.max_pool2d(F.relu(self.conv1(x)), (2, 2))
        # If the size is a square you can only specify a single number
        x = F.max_pool2d(F.relu(self.conv2(x)), 2)
        x = x.view(-1, self.num_flat_features(x))
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

Define modules

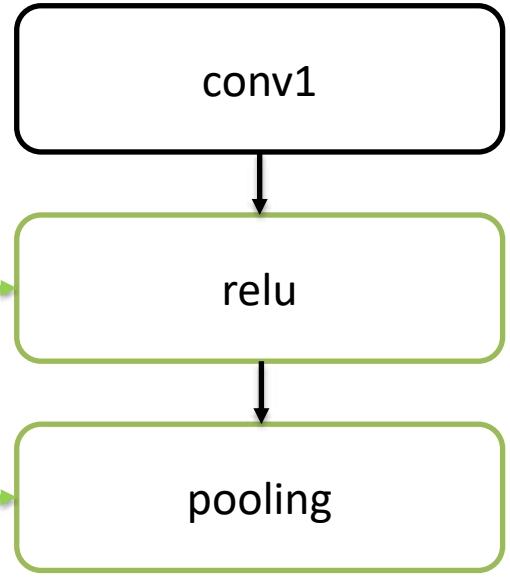
Build network

```python
import torch
import torch.nn as nn
import torch.nn.functional as F


class Net(nn.Module):

    def __init__(self):
        super(Net, self).__init__()
        # 1 input image channel, 6 output channels, 3x3 square convolution
        # kernel
        self.conv1 = nn.Conv2d(1, 6, 3)
        self.conv2 = nn.Conv2d(6, 16, 3)
        # an affine operation: y = Wx + b
        self.fc1 = nn.Linear(16 * 6 * 6, 120)  # 6*6 from image dimension
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        # Max pooling over a (2, 2) window
        x = F.max_pool2d(F.relu(self.conv1(x)), (2, 2))
        # If the size is a square you can only specify a single number
        x = F.max_pool2d(F.relu(self.conv2(x)), 2)
        x = x.view(-1, self.num_flat_features(x))
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```



```
import torch.optim as optim

criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)
```

```
net = Net()
```

Init Network

```
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()
```

Parameter update

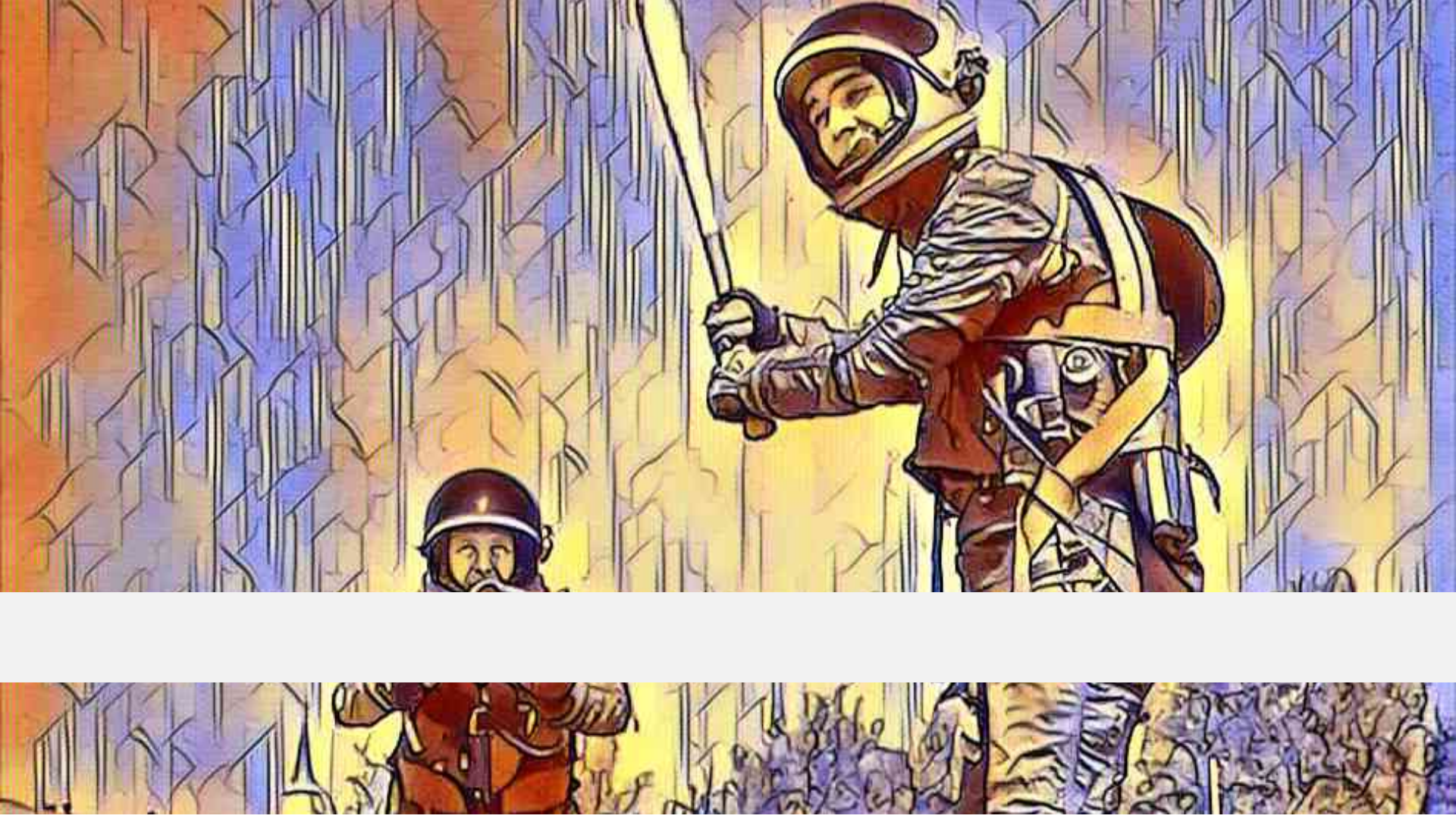
```
        # forward + backward + optimize
        outputs = net(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
```

Train

```
        # 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')
```

Monitor Progress



KÖNIGSWEG

Packages

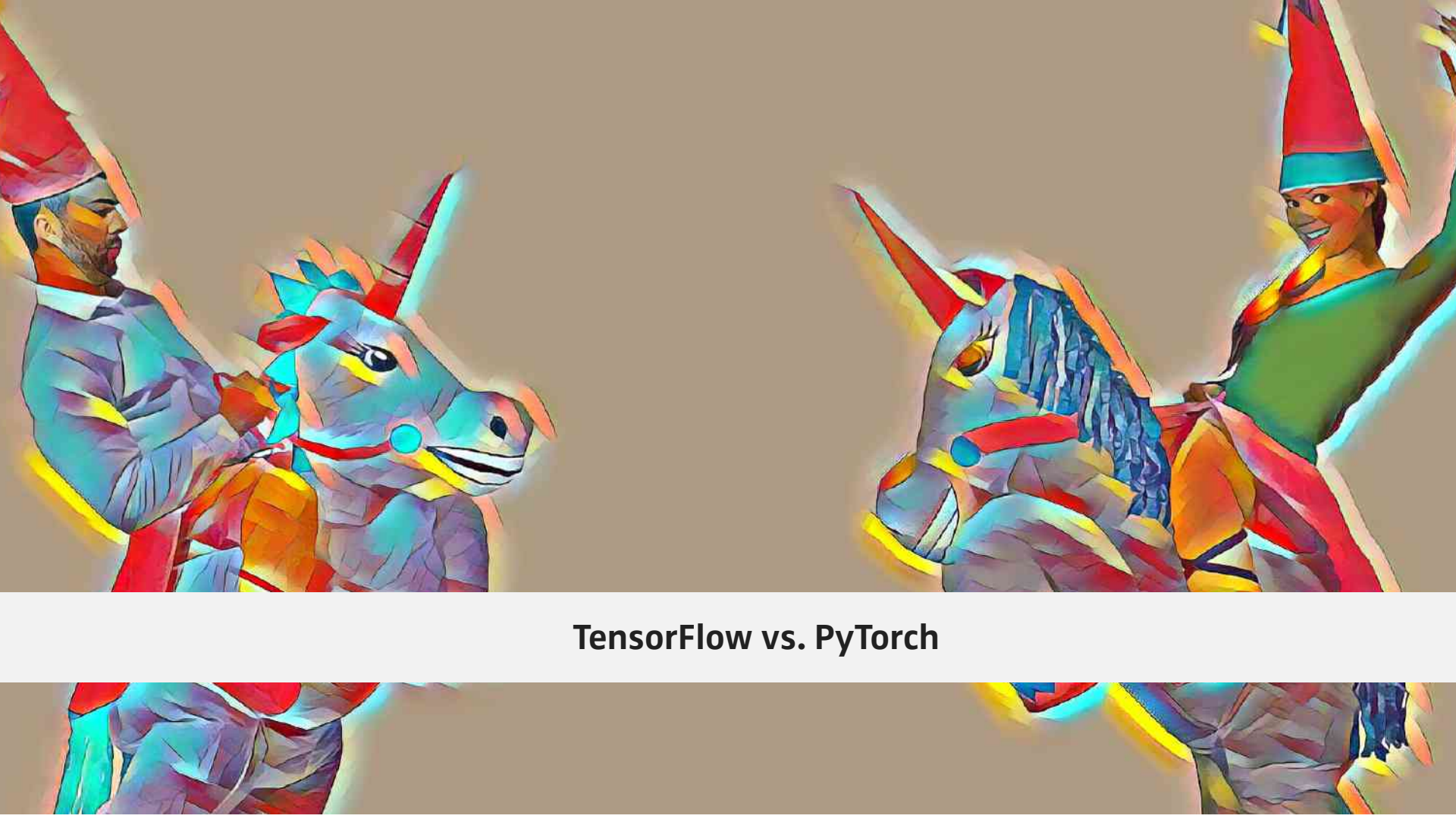
— torch	a Tensor library like Numpy, with strong GPU support
— torch.autograd	a tape based automatic differentiation library that supports all differentiable Tensor operations in torch
— torch.nn	a neural networks library deeply integrated with autograd designed for maximum flexibility
— torch.optim	an optimization package to be used with torch.nn with standard optimization methods such as SGD, RMSProp, LBFGS, Adam etc.
— torch.multiprocessing	python multiprocessing, but with magical memory sharing of torch Tensors across processes. Useful for data loading and hogwild training.
— torch.utils	DataLoader, Trainer and other utility functions for convenience

GPU / CPU

- Well integrated with CUDA
- Control where to put the data CPU / GPU
  `torch.device(dev)`
- Release data from GPU
  `torch.cuda.empty_cache()`
- `torch.cuda.is_available()`

Saving Models

- Pickle
- torch.save(the_model.state_dict(), optimizer_state_dict, PATH)
- Requires the code to load 😨
- ONNX



TensorFlow vs. PyTorch

Comparison with TensorFlow

Properties	TensorFlow	PyTorch
Graph	Static Dynamic (TensorFlow Fold)	Dynamic
Ramp-up Time	-	Win
Graph Creation and Debugging	-	Win
Feature Coverage	Win	Catch up quickly
Documentation	Tie	Tie
Serialization	Win (support other lang.)	-
Deployment	Win (Cloud & Mobile)	-
Data Loading	-	Win
Device Management	Win	Need .cuda()
Custom Extensions	-	Win

Summarized from https://awni.github.io/pytorch-tensorflow/

Wrap Up

KÖNIGSWEG

KÖNIGSWEG

Vielen Dank für Ihre Aufmerksamkeit. Fragen & Antworten

koenigsweg.com

ah@koenigsweg.com

@hendorf

KÖNIGSWEG

Kontakt

Königsweg GmbH

Musikpark Mannheim
Hafenstraße 49
68159 Mannheim

Mafinex Technologiezentrum
Julius-Hatry-Straße 1
68163 Mannheim

Telefon: +49 621 43 74 10 22
Telefax: +49 621 43 74 10 25

E-Mail: info@koenigsweg.com
Web: www.koenigsweg.com