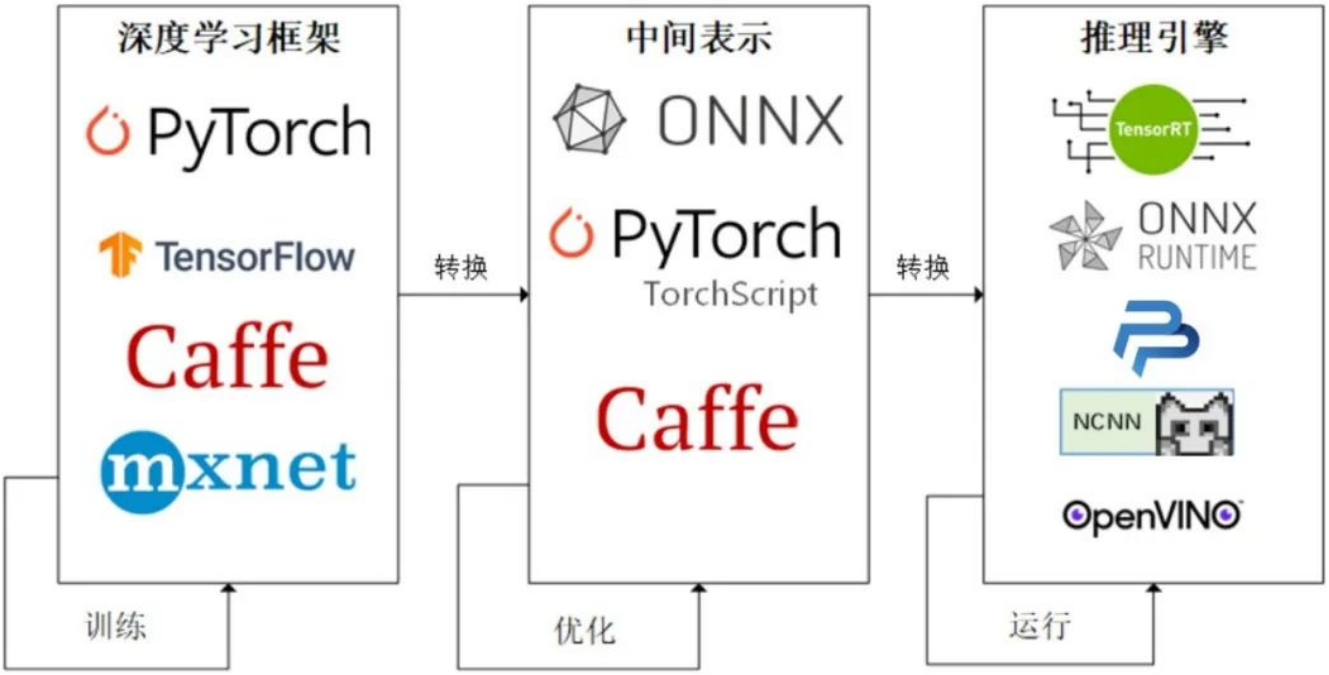
Note Information

Author : AbbasXu
Date : 2022-08-28
Title : pytorch基础
Keywords : #pytroch #模型部署

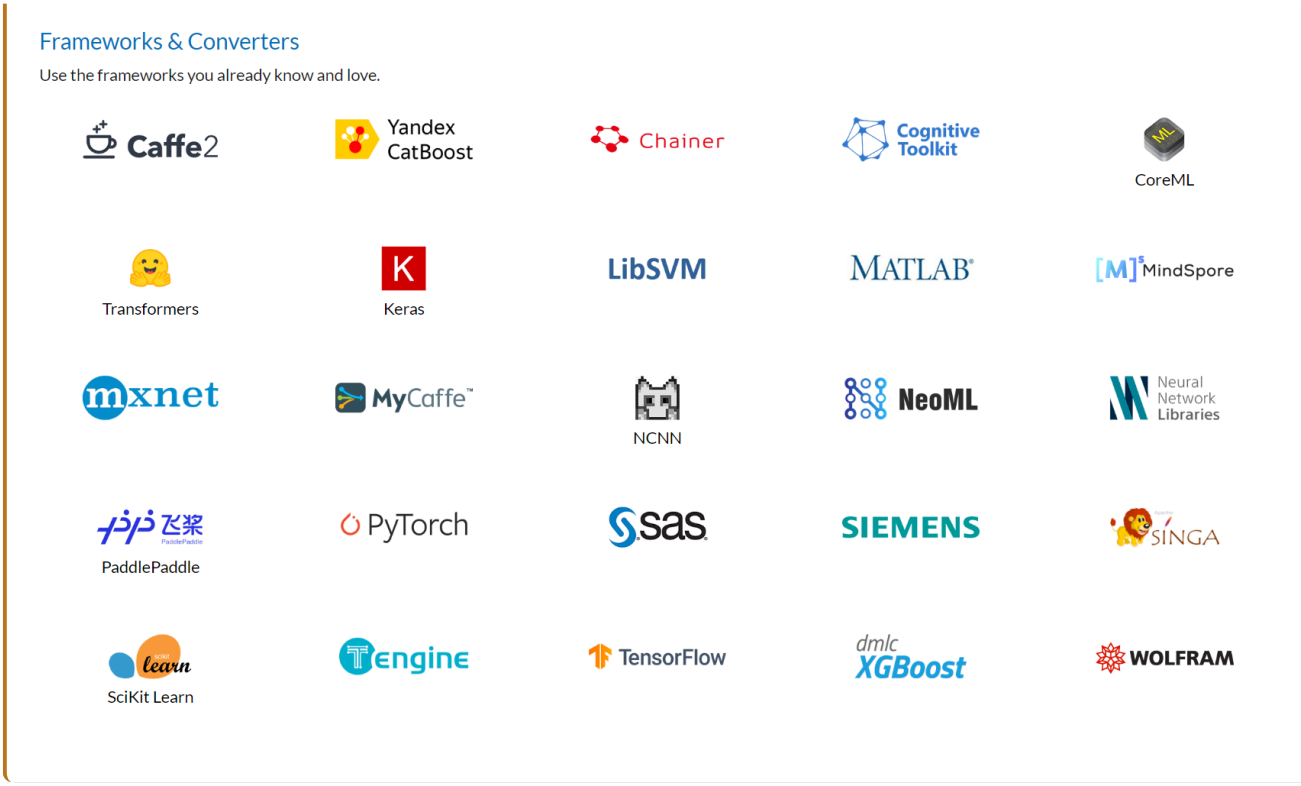
使用ONNX进行部署并推理

模型部署pipeline



ONNX简介

- ONNX官网：https://onnx.ai/
- ONNX GitHub：https://github.com/onnx/onnx
  ONNX(Open Neural Network Exchange) 是 Facebook (现Meta) 和微软在2017年共同发布的，用于标准描述计算图的一种格式。
  ONNX 已经对接了下图的多种深度学习框架和多种推理引擎。

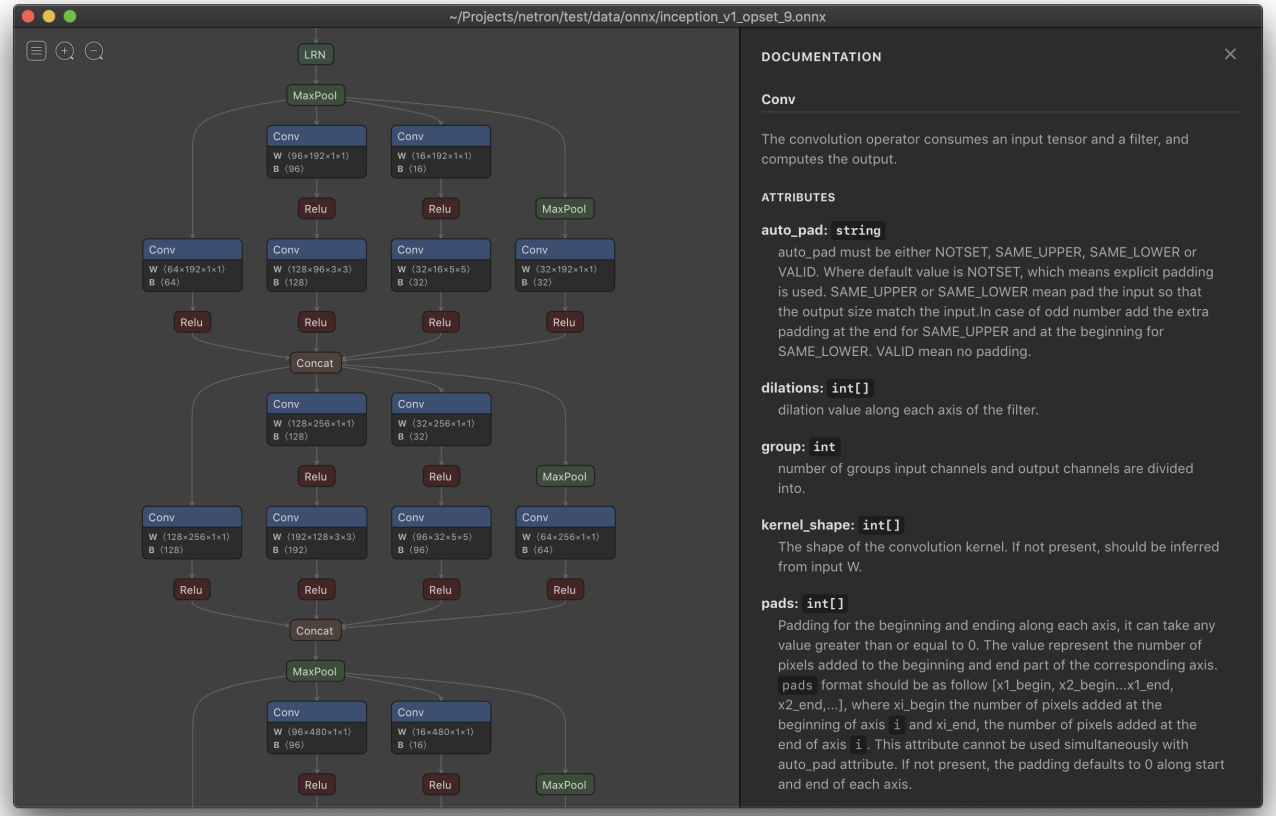


ONNX Runtime简介

- ONNX Runtime官网：https://www.onnxruntime.ai/
- ONNX Runtime GitHub：https://github.com/microsoft/onnxruntime
  ONNX Runtime 是由微软维护的一个跨平台机器学习推理加速器，它直接对接ONNX，可以直接读取.onnx文件并实现推理，不需要再把 .onnx 格式的文件转换成其他格式的文件。PyTorch借助ONNX Runtime也完成了部署的最后一公里，构建了 PyTorch --> ONNX --> ONNX Runtime 部署流水线，我们只需要将模型转换为 .onnx 文件，并在 ONNX Runtime 上运行模型即可。

模型导出为ONNX

使用`torch.onnx.export()`把模型转换成 ONNX 格式的函数。
通过`onnx.checker.check_model()`进行检验。
使用Netron ONNX可视化


~/Projects/netron/test/data/onnx/inception_v1_opset_9.onnx
LRN
MaxPool
Conv
W (96×192×1×1)
B (96)
Conv
W (16×192×1×1)
B (16)
Relu
Relu
MaxPool
Conv
W (64×192×1×1)
B (64)
Conv
W (128×96×3×3)
B (128)
Conv
W (32×16×5×5)
B (32)
Conv
W (32×192×1×1)
B (32)
Relu
Relu
Relu
Relu
Concat
Conv
W (128×256×1×1)
B (128)
Conv
W (32×256×1×1)
B (32)
Relu
Relu
MaxPool
Conv
W (128×256×1×1)
B (128)
Conv
W (192×128×3×3)
B (192)
Conv
W (96×32×5×5)
B (96)
Conv
W (64×256×1×1)
B (64)
Relu
Relu
Relu
Relu
Concat
MaxPool
Conv
W (96×480×1×1)
B (96)
Conv
W (16×480×1×1)
B (16)
Relu
Relu
MaxPool
DOCUMENTATION
Conv
The convolution operator consumes an input tensor and a filter, and computes the output.
ATTRIBUTES
auto_pad: string
auto_pad must be either NOTSET, SAME_UPPER, SAME_LOWER or VALID. Where default value is NOTSET, which means explicit padding is used. SAME_UPPER or SAME_LOWER mean pad the input so that the output size match the input.In case of odd number add the extra padding at the end for SAME_UPPER and at the beginning for SAME_LOWER. VALID mean no padding.
dilations: int[]
dilation value along each axis of the filter.
group: int
number of groups input channels and output channels are divided into.
kernel_shape: int[]
The shape of the convolution kernel. If not present, should be inferred from input W.
pads: int[]
Padding for the beginning and ending along each axis, it can take any value greater than or equal to 0. The value represent the number of pixels added to the beginning and end part of the corresponding axis. pads format should be as follow [x1_begin, x2_begin...x1_end, x2_end,...], where xi_begin the number of pixels added at the beginning of axis i and xi_end, the number of pixels added at the end of axis i. This attribute cannot be used simultaneously with auto_pad attribute. If not present, the padding defaults to 0 along start and end of each axis.