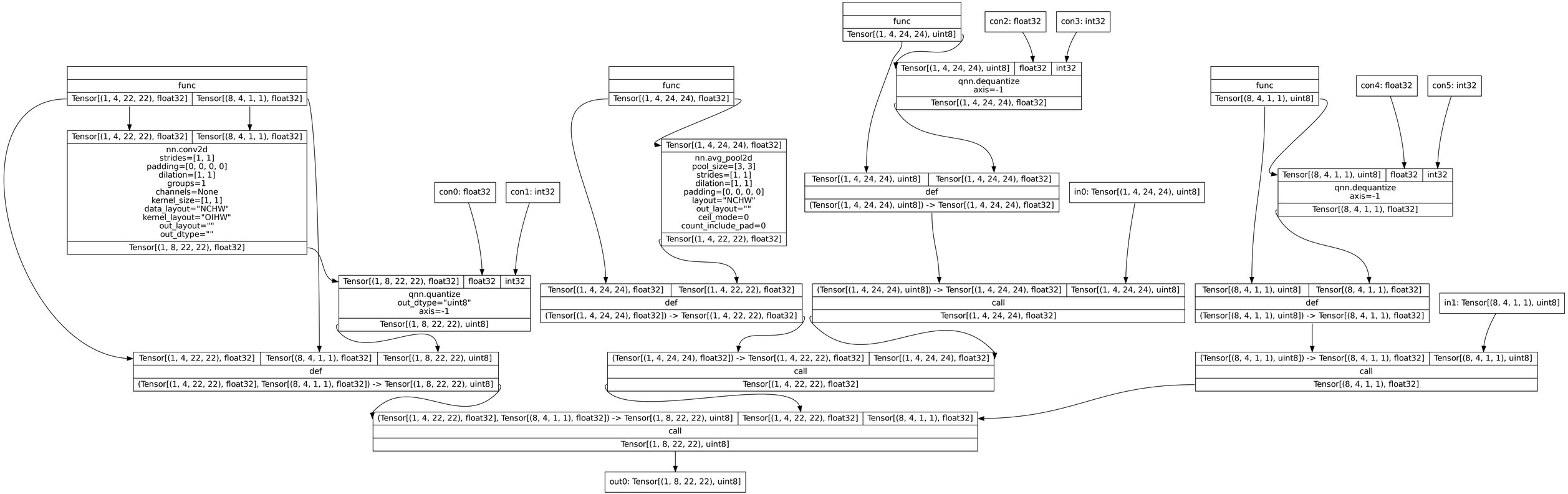
func
Tensor[(1, 4, 22, 22), float32]
Tensor[(8, 4, 1, 1), float32]
Tensor[(1, 4, 22, 22), float32]
Tensor[(8, 4, 1, 1), float32]
nn.conv2d
strides=[1, 1]
padding=[0, 0, 0, 0]
dilation=[1, 1]
groups=1
channels=None
kernel_size=[1, 1]
data_layout="NCHW"
kernel_layout="OIHW"
out_layout=""
out_dtype=""
Tensor[(1, 8, 22, 22), float32]
con0: float32
con1: int32
Tensor[(1, 8, 22, 22), float32]
float32
int32
qnn.quantize
out_dtype="uint8"
axis=-1
Tensor[(1, 8, 22, 22), uint8]
Tensor[(1, 4, 22, 22), float32]
Tensor[(8, 4, 1, 1), float32]
Tensor[(1, 8, 22, 22), uint8]
def
(Tensor[(1, 4, 22, 22), float32], Tensor[(8, 4, 1, 1), float32]) -> Tensor[(1, 8, 22, 22), uint8]
func
Tensor[(1, 4, 24, 24), float32]
Tensor[(1, 4, 24, 24), float32]
nn.avg_pool2d
pool_size=[3, 3]
strides=[1, 1]
dilation=[1, 1]
padding=[0, 0, 0, 0]
layout="NCHW"
out_layout=""
ceil_mode=0
count_include_pad=0
Tensor[(1, 4, 22, 22), float32]
Tensor[(1, 4, 24, 24), float32]
Tensor[(1, 4, 22, 22), float32]
def
(Tensor[(1, 4, 24, 24), float32]) -> Tensor[(1, 4, 22, 22), float32]
func
Tensor[(1, 4, 24, 24), uint8]
con2: float32
con3: int32
Tensor[(1, 4, 24, 24), uint8]
float32
int32
qnn.dequantize
axis=-1
Tensor[(1, 4, 24, 24), float32]
Tensor[(1, 4, 24, 24), uint8]
Tensor[(1, 4, 24, 24), float32]
def
(Tensor[(1, 4, 24, 24), uint8]) -> Tensor[(1, 4, 24, 24), float32]
in0: Tensor[(1, 4, 24, 24), uint8]
(Tensor[(1, 4, 24, 24), uint8]) -> Tensor[(1, 4, 24, 24), float32]
Tensor[(1, 4, 24, 24), uint8]
call
Tensor[(1, 4, 24, 24), float32]
(Tensor[(1, 4, 24, 24), float32]) -> Tensor[(1, 4, 22, 22), float32]
Tensor[(1, 4, 24, 24), float32]
call
Tensor[(1, 4, 22, 22), float32]
func
Tensor[(8, 4, 1, 1), uint8]
con4: float32
con5: int32
Tensor[(8, 4, 1, 1), uint8]
float32
int32
qnn.dequantize
axis=-1
Tensor[(8, 4, 1, 1), float32]
Tensor[(8, 4, 1, 1), uint8]
Tensor[(8, 4, 1, 1), float32]
def
(Tensor[(8, 4, 1, 1), uint8]) -> Tensor[(8, 4, 1, 1), float32]
in1: Tensor[(8, 4, 1, 1), uint8]
(Tensor[(8, 4, 1, 1), uint8]) -> Tensor[(8, 4, 1, 1), float32]
Tensor[(8, 4, 1, 1), uint8]
call
Tensor[(8, 4, 1, 1), float32]
(Tensor[(1, 4, 22, 22), float32], Tensor[(8, 4, 1, 1), float32]) -> Tensor[(1, 8, 22, 22), uint8]
Tensor[(1, 4, 22, 22), float32]
Tensor[(8, 4, 1, 1), float32]
call
Tensor[(1, 8, 22, 22), uint8]
out0: Tensor[(1, 8, 22, 22), uint8]