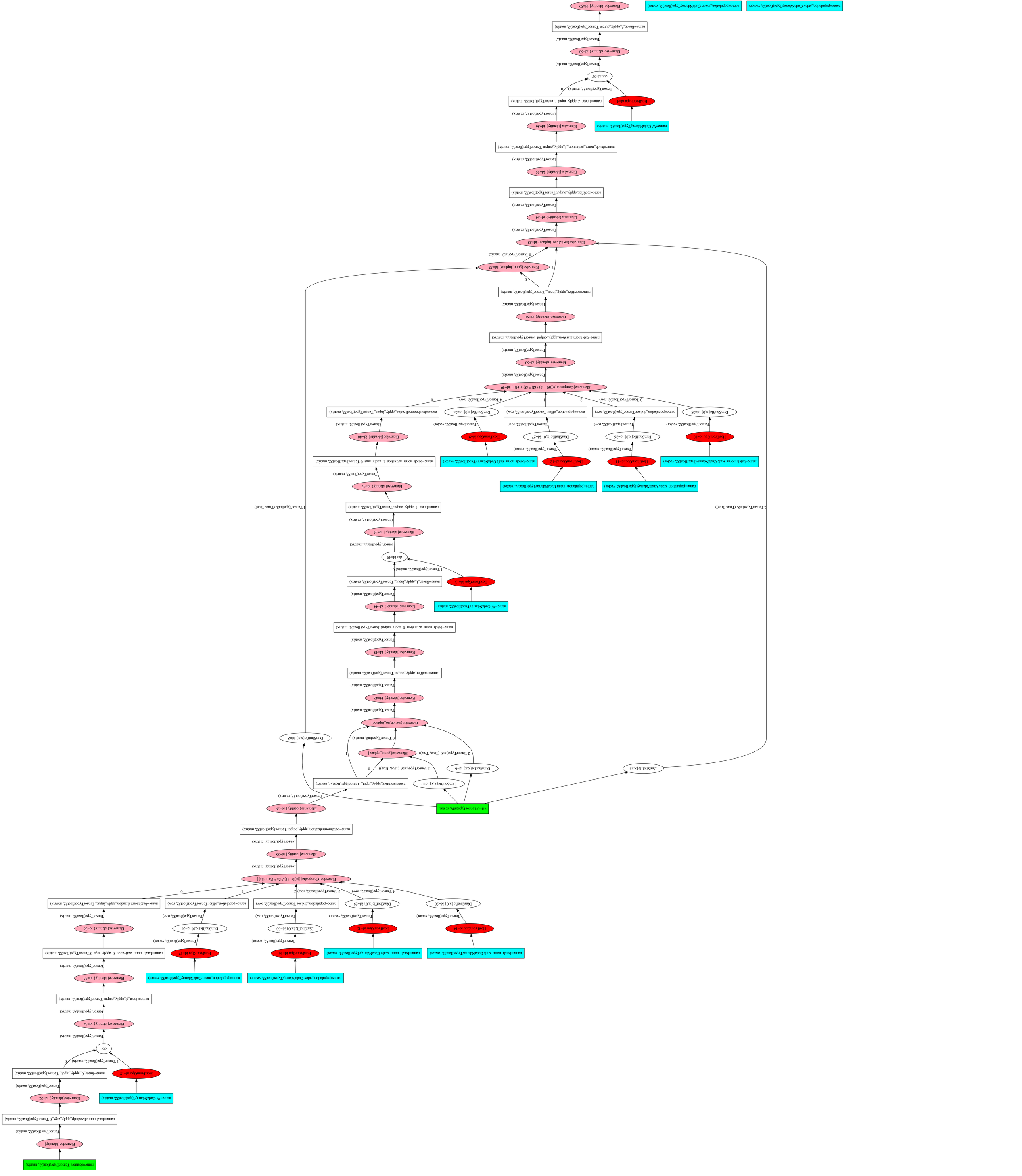
name=feature Tensor(float32, matrix)
Elemwise{identity}
Tensor(float32, matrix)
name=batchnormalizedmlp_apply_0 Tensor(float32, matrix)
Elemwise{identity} id=32
Tensor(float32, matrix)
name=W CudaNdarray(float32, matrix)
HostFromGpu id=18
name=linear_0_apply_input_ Tensor(float32, matrix)
1 Tensor(float32, matrix) 0
dot
Tensor(float32, matrix)
Elemwise{identity} id=34
Tensor(float32, matrix)
name=linear_0_apply_output Tensor(float32, matrix)
Elemwise{identity} id=35
Tensor(float32, matrix)
name=batch_norm_activation_0_apply_ags_0 Tensor(float32, matrix)
HostFromGpu id=17
Tensor(float32, vector)
Elemwise{identity} id=36
Tensor(float32, matrix)
name=batchnormalization_apply_input_ Tensor(float32, matrix)
name=population_mean CudaNdarray(float32, vector)
name=population_stder CudaNdarray(float32, vector)
HostFromGpu id=16
Tensor(float32, vector)
DimShuffle{x,0} id=31
Tensor(float32, row)
DimShuffle{x,0} id=30
Tensor(float32, row)
name=population_offset Tensor(float32, row)
name=population_divisor Tensor(float32, row)
name=batch_norm_shift CudaNdarray(float32, vector)
name=batch_norm_scale CudaNdarray(float32, vector)
HostFromGpu id=15
Tensor(float32, vector)
DimShuffle{x,0} id=29
Tensor(float32, vector)
DimShuffle{x,0} id=28
HostFromGpu id=14
4 Tensor(float32, row) 3 Tensor(float32, row) 2 1 0
Elemwise{Composite[(((((i0 - i1) / i2) * i3) + i4))]}
Tensor(float32, matrix)
Elemwise{identity} id=38
Tensor(float32, matrix)
name=batchnormalization_apply_output Tensor(float32, matrix)
Elemwise{identity} id=39
Tensor(float32, matrix)
Elemwise{identity}
Tensor(float32, matrix)
name=linear_0_apply_output Tensor(float32, matrix)
Elemwise{identity} id=40
Tensor(float32, matrix)
name=batch_norm_activation_0_apply_ags_0 Tensor(float32, matrix)
Elemwise{identity} id=41
Tensor(float32, matrix)
name=linear_1_apply_input_ Tensor(float32, matrix)
HostFromGpu id=13
1 Tensor(float32, matrix) 0
dot id=45
Tensor(float32, matrix)
Elemwise{identity} id=46
Tensor(float32, matrix)
name=linear_1_apply_output Tensor(float32, matrix)
Elemwise{identity} id=47
Tensor(float32, matrix)
name=batch_norm_activation_1_apply_ags_0 Tensor(float32, matrix)
Elemwise{identity} id=48
Tensor(float32, matrix)
name=population_mean CudaNdarray(float32, vector)
name=population_stder CudaNdarray(float32, vector)
HostFromGpu id=12
Tensor(float32, vector)
HostFromGpu id=11
Tensor(float32, vector)
HostFromGpu id=10
Tensor(float32, vector)
DimShuffle{x,0} id=27
Tensor(float32, row)
DimShuffle{x,0} id=26
Tensor(float32, row)
DimShuffle{x,0} id=25
Tensor(float32, vector)
HostFromGpu id=9
Tensor(float32, vector)
DimShuffle{x,0} id=24
Tensor(float32, row)
name=population_offset Tensor(float32, row)
name=population_divisor Tensor(float32, row)
name=batchnormalization_apply_input_ Tensor(float32, matrix)
4 Tensor(float32, row) 3 Tensor(float32, row) 2 1 0
Elemwise{Composite[(((((i0 - i1) / i2) * i3) + i4))]}
Tensor(float32, matrix)
Elemwise{identity} id=49
Tensor(float32, matrix)
name=linear_2_apply_output Tensor(float32, matrix)
Elemwise{identity} id=50
Tensor(float32, matrix)
name=batchnormalization_apply_output Tensor(float32, matrix)
Elemwise{identity} id=51
Tensor(float32, matrix)
name=rectifier_apply_input_ Tensor(float32, matrix)
Elemwise{gt,no_inplace} id=52
0 Tensor(int8, matrix)
Elemwise{switch,no_inplace} id=53
Tensor(float32, matrix)
Elemwise{identity} id=54
Tensor(float32, matrix)
name=rectifier_apply_output Tensor(float32, matrix)
Elemwise{identity} id=55
Tensor(float32, matrix)
name=batch_norm_activation_1_apply_output Tensor(float32, matrix)
Elemwise{identity} id=56
Tensor(float32, matrix)
HostFromGpu id=4
1 Tensor(float32, matrix) 0
dot id=57
Tensor(float32, matrix)
Elemwise{identity} id=58
Tensor(float32, matrix)
name=linear_2_apply_output Tensor(float32, matrix)
Elemwise{identity} id=59
Tensor(float32, matrix)
name=population_mean CudaNdarray(float32, vector)
name=population_stder CudaNdarray(float32, vector)
HostFromGpu id=5
Tensor(float32, vector)
HostFromGpu id=2
Tensor(float32, vector)
DimShuffle{x,0} id=23
Tensor(float32, row)
DimShuffle{x,0} id=22
Tensor(float32, row)
name=population_offset Tensor(float32, row)
name=population_divisor Tensor(float32, row)
name=batchnormalization_apply_output Tensor(float32, matrix)
Elemwise{identity} id=60
Tensor(float32, matrix)
name=batch_norm_activation_2_apply_ags_0 Tensor(float32, matrix)
Elemwise{identity} id=61
Tensor(float32, matrix)
name=batchnormalization_apply_output Tensor(float32, matrix)
Elemwise{identity} id=62
Tensor(float32, matrix)
name=batchnormalization_apply_output Tensor(float32, matrix)
Elemwise{identity} id=63
Tensor(float32, matrix)
name=softmax_apply_input_ Tensor(float32, matrix)
Softmax
Tensor(float32, matrix)
Elemwise{identity} id=65
Tensor(float32, matrix)
name=softmax_apply_output Tensor(float32, matrix)
Elemwise{identity} id=66
Tensor(float32, matrix)
name=batch_norm_activation_2_apply_output Tensor(float32, matrix)
Elemwise{identity} id=67
name=batchnormalizedmlp_apply_output Tensor(float32, matrix)
DimShuffle{x,0}
Tensor(float32, vector)
HostFromGpu id=1
Tensor(float32, vector)
DimShuffle{x,0} id=21
Tensor(float32, row)
name=population_divisor Tensor(float32, row)
name=population_offset Tensor(float32, row)
name=batchnormalization_apply_input_ Tensor(float32, matrix)
4 Tensor(float32, row) 3 Tensor(float32, row) 2 1 0
Elemwise{Composite[(((((i0 - i1) / i2) * i3) + i4))]}
Tensor(float32, matrix)
Elemwise{identity} id=64
Tensor(float32, matrix)
name=batchnormalization_apply_output Tensor(float32, matrix)
Elemwise{identity} id=63
Tensor(float32, matrix)
name=softmax_apply_input_ Tensor(float32, matrix)
Softmax
Tensor(float32, matrix)
Elemwise{identity} id=65
Tensor(float32, matrix)
name=softmax_apply_output Tensor(float32, matrix)
Elemwise{identity} id=66
Tensor(float32, matrix)
name=batch_norm_activation_2_apply_output Tensor(float32, matrix)
Elemwise{identity} id=67
name=batchnormalizedmlp_apply_output Tensor(float32, matrix)
2 Tensor(int8, (True, True))
1 Tensor(int8, (True, True))