

onnxruntime_training_nb

January 17, 2023

1 Stochastic Gradient Descent on simple function

`onnxruntime-training` is an extension `onnxruntime` or more precisely the same library compiled with different settings. It provides a way to compute a gradient of a function defined by an ONNX graph.

```
[1]: from jyquickhelper import add_notebook_menu
      add_notebook_menu()
```

```
[1]: <IPython.core.display.HTML object>
```

```
[2]: %load_ext mlproduct
```

1.1 A simple problem

Let's choose a simple regression problem defined by $z = -1 - 2x + 3y + \frac{1}{2}x^2 - \frac{1}{3}y^2 + \epsilon$ and we try to approximate by a function $f(x, y) = a + bx + cy + dx^2 + ey^2$. Every coefficient is determined from an optimization problem solved with a stochastic gradient descent.

```
[3]: from typing import Any
      import numpy
      import mlproduct.npy.numpy_onnx_impl as npnx
      from mlproduct.npy import onnxnumpy_default, NDArray

      @onnxnumpy_default
      def fct(x: NDArray[(None, 2), numpy.float32]) -> NDArray[(None, 1), numpy.float32]:
          coef_x = numpy.array([[-2, 3]], dtype=numpy.float32)
          coef_x2 = numpy.array([[0.5, -0.33333]], dtype=numpy.float32)
          bias = numpy.array([-1], dtype=numpy.float32)
          poly = x * coef_x + x * x * coef_x2
          y = poly[:, 0] + poly[:, 1] + bias
          return y.reshape((-1, 1))

      x = numpy.array([[0, 0], [1, 0], [0, 1], [1, 1], [2, 2]], dtype=numpy.float32)
      fct(x)
```

```
[3]: array([[-1.          ],
            [-2.5         ],
            [ 1.6666701   ],
            [ 0.16667008  ],
            [ 1.6666799   ]], dtype=float32)
```

```
[4]: %onnxview fct.to_onnx()
```

```
[4]: <jyquickhelper.jspy.render_nb_js_dot.RenderJsDot at 0x162913e2fa0>
```

```
[5]: from mlproduct.plotting.text_plot import onnx_simple_text_plot
print(onnx_simple_text_plot(fct.to_onnx()))
```

```
opset: domain='' version=14
input: name='x' type=dtype('float32') shape=(0, 2)
init: name='init' type=dtype('float32') shape=(0,) -- array([ 0.5      ,
-0.33333], dtype=float32)
init: name='init_1' type=dtype('float32') shape=(0,) -- array([-2.,  3.],
dtype=float32)
init: name='init_2' type=dtype('int64') shape=(0,) -- array([1], dtype=int64)
init: name='init_3' type=dtype('int64') shape=(0,) -- array([2], dtype=int64)
init: name='init_5' type=dtype('int64') shape=(0,) -- array([0], dtype=int64)
init: name='init_b10' type=dtype('float32') shape=(0,) -- array([-1.],
dtype=float32)
init: name='init_b11' type=dtype('int64') shape=(0,) -- array([-1,  1],
dtype=int64)
Mul(x, x) -> out_mul_0
Mul(out_mul_0, init) -> out_mul_0_1
Mul(x, init_1) -> out_mul_0_2
Add(out_mul_0_2, out_mul_0_1) -> out_add_0
Slice(out_add_0, init_2, init_3, init_2) -> out_sli_0
Squeeze(out_sli_0, init_2) -> out_squ_0
Slice(out_add_0, init_5, init_2, init_2) -> out_sli_0_1
Squeeze(out_sli_0_1, init_2) -> out_squ_0_1
Add(out_squ_0_1, out_squ_0) -> out_add_0_1
Add(out_add_0_1, init_b10) -> out_add_0_2
Reshape(out_add_0_2, init_b11) -> y
output: name='y' type=dtype('float32') shape=(0, 1)
```

1.2 Gradient : retropropagation

Let's look into the gradient.

```
[6]: from onnxcustom.training.grad_helper import onnx_derivative, DerivativeOptions

onx = fct.to_onnx()
grad = onnx_derivative(onx)
%onnxview grad
```

```
[6]: <jyquickhelper.jspy.render_nb_js_dot.RenderJsDot at 0x162913e79d0>
```

```
[7]: from mlproduct.plotting.text_plot import onnx_text_plot_io, onnx_simple_text_plot
print(onnx_text_plot_io(grad))
```

```
opset: domain='' version=14
opset: domain='com.microsoft.nchwc' version=1
opset: domain='ai.onnx.ml' version=2
opset: domain='com.ms.internal.nhwc' version=1
opset: domain='ai.onnx.training' version=1
```

```

opset: domain='ai.onnx.preview.training' version=1
opset: domain='com.microsoft' version=1
opset: domain='com.microsoft.experimental' version=1
input: name='x' type=dtype('float32') shape=(0, 2)
input: name='init' type=dtype('float32') shape=(1, 2)
input: name='init_1' type=dtype('float32') shape=(1, 2)
input: name='init_b10' type=dtype('float32') shape=(1,)
input: name='y_grad' type=dtype('float32') shape=(0, 1)
init: name='init_5' type=dtype('int64') shape=(0,)
init: name='init_2' type=dtype('int64') shape=(0,)
init: name='init_3' type=dtype('int64') shape=(0,)
output: name='x_grad' type=dtype('float32') shape=(0, 2)
output: name='init_grad' type=dtype('float32') shape=(1, 2)
output: name='init_1_grad' type=dtype('float32') shape=(1, 2)
output: name='init_b10_grad' type=dtype('float32') shape=(1,)

```

```

[8]: from mlproduct.onnx_tools.onnx_manipulations import onnx_rename_names
      renamed = onnx_rename_names(grad)

```

```

[9]: print(onnx_simple_text_plot(renamed))

```

```

opset: domain='' version=14
opset: domain='com.microsoft.nchwc' version=1
opset: domain='ai.onnx.ml' version=2
opset: domain='com.ms.internal.nhwc' version=1
opset: domain='ai.onnx.training' version=1
opset: domain='ai.onnx.preview.training' version=1
opset: domain='com.microsoft' version=1
opset: domain='com.microsoft.experimental' version=1
input: name='x' type=dtype('float32') shape=(0, 2)
input: name='init' type=dtype('float32') shape=(1, 2)
input: name='init_1' type=dtype('float32') shape=(1, 2)
input: name='init_b10' type=dtype('float32') shape=(1,)
input: name='y_grad' type=dtype('float32') shape=(0, 1)
init: name='i0' type=dtype('int64') shape=(0,) -- array([0], dtype=int64)
init: name='i1' type=dtype('int64') shape=(0,) -- array([1], dtype=int64)
init: name='i2' type=dtype('int64') shape=(0,) -- array([2], dtype=int64)
Mul(x, x) -> r0
  Mul(r0, init) -> r1
    Shape(r1) -> r32
Mul(x, init_1) -> r2
  Add(r2, r1) -> r3
    Slice(r3, i1, i2, i1) -> r4
      Squeeze(r4, i1) -> r5
        Shape(r5) -> r18
      Slice(r3, i0, i1, i1) -> r6
        Squeeze(r6, i1) -> r7
          Add(r7, r5) -> r8
            Add(r8, init_b10) -> r9
              Shape(r9) -> r10
                Reshape(y_grad, r10, allowzero=0) -> r11
              Shape(init_b10) -> r12
            Shape(r8) -> r13

```

```

BroadcastGradientArgs(r13, r12) -> r14, r15
  ReduceSum(r11, r14, keepdims=1, noop_with_empty_axes=1) -> r16
  Reshape(r16, r13, allowzero=0) -> r17
Shape(r7) -> r19
BroadcastGradientArgs(r19, r18) -> r20, r21
  ReduceSum(r17, r21, keepdims=1, noop_with_empty_axes=1) -> r22
  Reshape(r22, r18, allowzero=0) -> r23
  Unsqueeze(r23, i1) -> r24
  Shape(r3) -> r25
  SliceGrad(r24, r25, i1, i2, i1) -> r26
  ReduceSum(r17, r20, keepdims=1, noop_with_empty_axes=1) -> r27
Reshape(r27, r19, allowzero=0) -> r28
  Unsqueeze(r28, i1) -> r29
  SliceGrad(r29, r25, i0, i1, i1) -> r30
  Sum(r30, r26) -> r31
Shape(r2) -> r33
  BroadcastGradientArgs(r33, r32) -> r34, r35
  ReduceSum(r31, r35, keepdims=1, noop_with_empty_axes=1) -> r36
  Reshape(r36, r32, allowzero=0) -> r37
  Mul(r37, init) -> r38
Shape(init) -> r39
Shape(r0) -> r40
  BroadcastGradientArgs(r40, r39) -> r41, r42
  ReduceSum(r38, r41, keepdims=1, noop_with_empty_axes=1) -> r43
  Reshape(r43, r40, allowzero=0) -> r44
  Mul(r44, x) -> r45
ReduceSum(r31, r34, keepdims=1, noop_with_empty_axes=1) -> r46
  Reshape(r46, r33, allowzero=0) -> r47
  Mul(r47, init_1) -> r48
Shape(init_1) -> r49
Shape(x) -> r50
  BroadcastGradientArgs(r50, r49) -> r51, r52
  ReduceSum(r48, r51, keepdims=1, noop_with_empty_axes=1) -> r53
  Reshape(r53, r50, allowzero=0) -> r54
  Sum(r54, r45, r45) -> x_grad
ReduceSum(r11, r15, keepdims=1, noop_with_empty_axes=1) -> r55
  Reshape(r55, r12, allowzero=0) -> init_b10_grad
Mul(r37, r0) -> r56
  ReduceSum(r56, r42, keepdims=1, noop_with_empty_axes=1) -> r57
  Reshape(r57, r39, allowzero=0) -> init_grad
Mul(r47, x) -> r58
  ReduceSum(r58, r52, keepdims=1, noop_with_empty_axes=1) -> r59
  Reshape(r59, r49, allowzero=0) -> init_1_grad
output: name='x_grad' type=dtype('float32') shape=(0, 2)
output: name='init_grad' type=dtype('float32') shape=(1, 2)
output: name='init_1_grad' type=dtype('float32') shape=(1, 2)
output: name='init_b10_grad' type=dtype('float32') shape=(1,)

```

```
[10]: set(n.op_type for n in grad.graph.node)
```

```
[10]: {'Add',
      'BroadcastGradientArgs',
      'Mul',
      'ReduceSum',

```

```

'Reshape',
'Shape',
'Slice',
'SliceGrad',
'Squeeze',
'Sum',
'Unsqueeze'}

```

The resulting graph assumes the gradient for `y_grad` is known. That's the case for a layer in a neural network. In our case, this gradient should come from the loss. Let's add it to the graph.

1.3 Add a square loss

```

[11]: from onnxcustom.utils.orttraining_helper import add_loss_output
      onx_loss = add_loss_output(onx)

      %onnxview onx_loss

```

```

[11]: <jyquickhelper.jspy.render_nb_js_dot.RenderJsDot at 0x162913e7640>

```

```

[12]: print(onnx_simple_text_plot(onx_loss))

```

```

opset: domain='' version=14
input: name='x' type=dtype('float32') shape=(0, 2)
input: name='label' type=dtype('float32') shape=(0, 1)
init: name='init' type=dtype('float32') shape=(0,) -- array([ 0.5      ,
-0.33333], dtype=float32)
init: name='init_1' type=dtype('float32') shape=(0,) -- array([-2.,  3.],
dtype=float32)
init: name='init_2' type=dtype('int64') shape=(0,) -- array([1], dtype=int64)
init: name='init_3' type=dtype('int64') shape=(0,) -- array([2], dtype=int64)
init: name='init_5' type=dtype('int64') shape=(0,) -- array([0], dtype=int64)
init: name='init_b10' type=dtype('float32') shape=(0,) -- array([-1.],
dtype=float32)
init: name='init_b11' type=dtype('int64') shape=(0,) -- array([-1,  1],
dtype=int64)
Mul(x, x) -> out_mul_0
  Mul(out_mul_0, init) -> out_mul_0_1
Mul(x, init_1) -> out_mul_0_2
  Add(out_mul_0_2, out_mul_0_1) -> out_add_0
    Slice(out_add_0, init_2, init_3, init_2) -> out_sli_0
      Squeeze(out_sli_0, init_2) -> out_squ_0
    Slice(out_add_0, init_5, init_2, init_2) -> out_sli_0_1
      Squeeze(out_sli_0_1, init_2) -> out_squ_0_1
    Add(out_squ_0_1, out_squ_0) -> out_add_0_1
      Add(out_add_0_1, init_b10) -> out_add_0_2
        Reshape(out_add_0_2, init_b11) -> y
          Sub(y, label) -> loss_diff
            Mul(loss_diff, loss_diff) -> loss_diff_2
              ReduceSum(loss_diff_2) -> loss
output: name='loss' type=dtype('float32') shape=(1, 1)
output: name='y' type=dtype('float32') shape=(0, 1)

```

The graph has 5 inputs: x, label or the expected target, and the weights and two outputs, the function output and the loss. We don't need the first one so we remove it.

```
[13]: from mlprodict.onnx_tools.onnx_manipulations import select_model_inputs_outputs
```

```
onx_loss_only = select_model_inputs_outputs(onx_loss, outputs=['loss'])
print(onnx_simple_text_plot(onx_loss_only))
```

```
opset: domain='' version=14
input: name='x' type=dtype('float32') shape=(0, 2)
input: name='label' type=dtype('float32') shape=(0, 1)
init: name='init' type=dtype('float32') shape=(0,) -- array([ 0.5      ,
-0.33333], dtype=float32)
init: name='init_1' type=dtype('float32') shape=(0,) -- array([-2.,  3.],
dtype=float32)
init: name='init_2' type=dtype('int64') shape=(0,) -- array([1], dtype=int64)
init: name='init_3' type=dtype('int64') shape=(0,) -- array([2], dtype=int64)
init: name='init_5' type=dtype('int64') shape=(0,) -- array([0], dtype=int64)
init: name='init_b10' type=dtype('float32') shape=(0,) -- array([-1.],
dtype=float32)
init: name='init_b11' type=dtype('int64') shape=(0,) -- array([-1,  1],
dtype=int64)
Mul(x, x) -> out_mul_0
  Mul(out_mul_0, init) -> out_mul_0_1
Mul(x, init_1) -> out_mul_0_2
  Add(out_mul_0_2, out_mul_0_1) -> out_add_0
    Slice(out_add_0, init_5, init_2, init_2) -> out_sli_0_1
      Squeeze(out_sli_0_1, init_2) -> out_squ_0_1
    Slice(out_add_0, init_2, init_3, init_2) -> out_sli_0
      Squeeze(out_sli_0, init_2) -> out_squ_0
    Add(out_squ_0_1, out_squ_0) -> out_add_0_1
      Add(out_add_0_1, init_b10) -> out_add_0_2
        Reshape(out_add_0_2, init_b11) -> y
          Sub(y, label) -> loss_diff
            Mul(loss_diff, loss_diff) -> loss_diff_2
              ReduceSum(loss_diff_2) -> loss
output: name='loss' type=dtype('float32') shape=(1, 1)
```

1.4 Gradient again : loss + retropropagation

```
[14]: grad_loss = onnx_rename_names(onnx_derivative(
    onx_loss_only, options=DerivativeOptions.FillGrad | DerivativeOptions.KeepOutputs))
%onnxview grad_loss
```

```
[14]: <jyquickhelper.jsipy.render_nb_js_dot.RenderJsDot at 0x162913e7d90>
```

```
[15]: print(onnx_simple_text_plot(grad_loss))
```

```
opset: domain='' version=14
opset: domain='com.microsoft.nchwc' version=1
opset: domain='ai.onnx.ml' version=2
opset: domain='com.ms.internal.nhwc' version=1
opset: domain='ai.onnx.training' version=1
```

```

opset: domain='ai.onnx.preview.training' version=1
opset: domain='com.microsoft' version=1
opset: domain='com.microsoft.experimental' version=1
input: name='x' type=dtype('float32') shape=(0, 2)
input: name='label' type=dtype('float32') shape=(0, 1)
input: name='init' type=dtype('float32') shape=(1, 2)
input: name='init_1' type=dtype('float32') shape=(1, 2)
input: name='init_b10' type=dtype('float32') shape=(1,)
init: name='i0' type=dtype('int64') shape=(0,) -- array([-1, 1], dtype=int64)
init: name='i1' type=dtype('int64') shape=(0,) -- array([0], dtype=int64)
init: name='i2' type=dtype('int64') shape=(0,) -- array([1], dtype=int64)
init: name='i3' type=dtype('int64') shape=(0,) -- array([2], dtype=int64)
Mul(x, init_1) -> r0
  Shape(r0) -> r47
Mul(x, x) -> r1
  Mul(r1, init) -> r2
  Add(r0, r2) -> r3
    Slice(r3, i1, i2, i2) -> r4
    Squeeze(r4, i2) -> r5
    Shape(r5) -> r33
    Slice(r3, i2, i3, i2) -> r6
    Squeeze(r6, i2) -> r7
    Add(r5, r7) -> r8
    Add(r8, init_b10) -> r9
    Reshape(r9, i0, allowzero=0) -> r10
    Sub(r10, label) -> r11
    Mul(r11, r11) -> r12
    ReduceSum(r12, keepdims=1, noop_with_empty_axes=0) -> loss
    Shape(loss) -> r76
    ConstantOfShape(r76) -> r14
    Shape(r12) -> r13
    Expand(r14, r13) -> r15
    Mul(r15, r11) -> r16
    Sum(r16, r16) -> r17
Shape(label) -> r18
Shape(r10) -> r19
  BroadcastGradientArgs(r19, r18) -> r20, r21
  ReduceSum(r17, r20, keepdims=1, noop_with_empty_axes=1) -> r22
  Reshape(r22, r19, allowzero=0) -> r23
Shape(r9) -> r24
  Reshape(r23, r24, allowzero=0) -> r25
Shape(init_b10) -> r26
Shape(r8) -> r27
  BroadcastGradientArgs(r27, r26) -> r28, r29
  ReduceSum(r25, r28, keepdims=1, noop_with_empty_axes=1) -> r30
  Reshape(r30, r27, allowzero=0) -> r31
Shape(r7) -> r32
  BroadcastGradientArgs(r33, r32) -> r34, r35
  ReduceSum(r31, r34, keepdims=1, noop_with_empty_axes=1) -> r36
  Reshape(r36, r33, allowzero=0) -> r37
  Unsqueeze(r37, i2) -> r38
  Shape(r3) -> r39
  SliceGrad(r38, r39, i1, i2, i2) -> r40
  ReduceSum(r31, r35, keepdims=1, noop_with_empty_axes=1) -> r41

```

```

Reshape(r41, r32, allowzero=0) -> r42
Unsqueeze(r42, i2) -> r43
SliceGrad(r43, r39, i2, i3, i2) -> r44
Sum(r44, r40) -> r45
Shape(r2) -> r46
BroadcastGradientArgs(r47, r46) -> r48, r49
ReduceSum(r45, r48, keepdims=1, noop_with_empty_axes=1) -> r50
Reshape(r50, r47, allowzero=0) -> r51
Mul(r51, init_1) -> r52
Shape(init_1) -> r53
Shape(x) -> r54
BroadcastGradientArgs(r54, r53) -> r55, r56
ReduceSum(r52, r55, keepdims=1, noop_with_empty_axes=1) -> r57
Reshape(r57, r54, allowzero=0) -> r58
ReduceSum(r45, r49, keepdims=1, noop_with_empty_axes=1) -> r59
Reshape(r59, r46, allowzero=0) -> r60
Mul(r60, init) -> r61
Shape(init) -> r62
Shape(r1) -> r63
BroadcastGradientArgs(r63, r62) -> r64, r65
ReduceSum(r61, r64, keepdims=1, noop_with_empty_axes=1) -> r66
Reshape(r66, r63, allowzero=0) -> r67
Mul(r67, x) -> r68
Sum(r68, r68, r58) -> x_grad
ReduceSum(r17, r21, keepdims=1, noop_with_empty_axes=1) -> r69
Reshape(r69, r18, allowzero=0) -> r70
Neg(r70) -> label_grad
ReduceSum(r25, r29, keepdims=1, noop_with_empty_axes=1) -> r71
Reshape(r71, r26, allowzero=0) -> init_b10_grad
Mul(r51, x) -> r72
ReduceSum(r72, r56, keepdims=1, noop_with_empty_axes=1) -> r73
Reshape(r73, r53, allowzero=0) -> init_1_grad
Mul(r60, r1) -> r74
ReduceSum(r74, r65, keepdims=1, noop_with_empty_axes=1) -> r75
Reshape(r75, r62, allowzero=0) -> init_grad
output: name='x_grad' type=dtype('float32') shape=(0, 2)
output: name='label_grad' type=dtype('float32') shape=(0, 1)
output: name='init_grad' type=dtype('float32') shape=(1, 2)
output: name='init_1_grad' type=dtype('float32') shape=(1, 2)
output: name='init_b10_grad' type=dtype('float32') shape=(1,)
output: name='loss' type=dtype('float32') shape=(1, 1)

```

Let's compute the gradient.

```
[16]: x
```

```
[16]: array([[0., 0.],
            [1., 0.],
            [0., 1.],
            [1., 1.],
            [2., 2.]], dtype=float32)
```

```
[17]: y = fct(x)
      y
```



```
[17]: array([[ -1.          ],
            [-2.5          ],
            [ 1.6666701   ],
            [ 0.16667008  ],
            [ 1.6666799   ]], dtype=float32)
```

```
[18]: from mlproduct.onnxrt import OnnxInference

oinf = OnnxInference(grad_loss, runtime='onnxruntime')
```

```
[19]: import pprint

init = numpy.array([[2, 3]], dtype=numpy.float32)
init_1 = numpy.array([[0.5, 0.33333]], dtype=numpy.float32)
init_b10 = numpy.array([1], dtype=numpy.float32)
result = oinf.run({'x': x, 'label': y,
                  'init': init, 'init_1': init_1, 'init_b10': init_b10})
pprint.pprint(result)
```

```
{'init_1_grad': array([[109.333244, 102.666565]], dtype=float32),
 'init_b10_grad': array([76.6666], dtype=float32),
 'init_grad': array([[193.33316, 186.66649]], dtype=float32),
 'label_grad': array([[ -4.          ],
                    [-12.          ],
                    [ -5.33332   ],
                    [-13.333321  ],
                    [-41.99996   ]], dtype=float32),
 'loss': array([[532.5546]], dtype=float32),
 'x_grad': array([[ 2.          ,  1.33332 ],
                 [ 54.          ,  3.99996 ],
                 [  2.66666   , 33.777676 ],
                 [ 59.999943  , 84.44432  ],
                 [356.99966   , 517.9994   ]], dtype=float32)}
```

We could use this gradient to implement a stochastic gradient descent in python. Two comments: * If we implement it this with numpy, it cannot work on GPU. * If we use OrtValue (tensor from onnxruntime), how to do simple addition between OrtValue ?

We need to implemented the second option. A simple addition between two OrtValue must be done with an ONNX graph.

1.5 TrainingSession

```
[20]: X = numpy.random.randn(100, 2).astype(numpy.float32) / 10
y = fct(X) + (numpy.random.randn(100, 1) / 1000).astype(numpy.float32)
X.shape, y.shape
```

```
[20]: ((100, 2), (100, 1))
```

```
[21]: print(onnx_simple_text_plot(onx))
```

```
opset: domain='' version=14
input: name='x' type=dtype('float32') shape=(0, 2)
init: name='init' type=dtype('float32') shape=(0,) -- array([ 0.5      ,
```

```

-0.33333], dtype=float32)
init: name='init_1' type=dtype('float32') shape=(0,) -- array([-2., 3.],
dtype=float32)
init: name='init_2' type=dtype('int64') shape=(0,) -- array([1], dtype=int64)
init: name='init_3' type=dtype('int64') shape=(0,) -- array([2], dtype=int64)
init: name='init_5' type=dtype('int64') shape=(0,) -- array([0], dtype=int64)
init: name='init_b10' type=dtype('float32') shape=(0,) -- array([-1.],
dtype=float32)
init: name='init_b11' type=dtype('int64') shape=(0,) -- array([-1, 1],
dtype=int64)
Mul(x, x) -> out_mul_0
Mul(out_mul_0, init) -> out_mul_0_1
Mul(x, init_1) -> out_mul_0_2
Add(out_mul_0_2, out_mul_0_1) -> out_add_0
Slice(out_add_0, init_2, init_3, init_2) -> out_sli_0
Squeeze(out_sli_0, init_2) -> out_squ_0
Slice(out_add_0, init_5, init_2, init_2) -> out_sli_0_1
Squeeze(out_sli_0_1, init_2) -> out_squ_0_1
Add(out_squ_0_1, out_squ_0) -> out_add_0_1
Add(out_add_0_1, init_b10) -> out_add_0_2
Reshape(out_add_0_2, init_b11) -> y
output: name='y' type=dtype('float32') shape=(0, 1)

```

```
[22]: from onnxcustom.training.optimizers import OrtGradientOptimizer
```

```

train_session = OrtGradientOptimizer(
    onx_loss, ['init', 'init_1', 'init_b10'], learning_rate=1e-1,
    batch_size=5, max_iter=100)

train_session.fit(X, y)

```

```

[22]: OrtGradientOptimizer(model_onnx='ir_version...', weights_to_train=['init',
'init_1', 'init_b10'], loss_output_name='loss', max_iter=100,
training_optimizer_name='SGDOptimizer', batch_size=5,
learning_rate=LearningRateSGD(eta0=0.1, alpha=0.0001, power_t=0.25,
learning_rate='invscaling'), value=0.03162277660168379, device='cpu',
warm_start=False, verbose=0, validation_every=10, saved_gradient=None,
sample_weight_name='weight')

```

```
[23]: train_session.trained_coef_
```

```

[23]: {'init': array([[ -0.34785354,  1.1399053 ]], dtype=float32),
'init_1': array([[ -1.9156165,  2.4292002 ]], dtype=float32),
'init_b10': array([ -1.0016667], dtype=float32)}

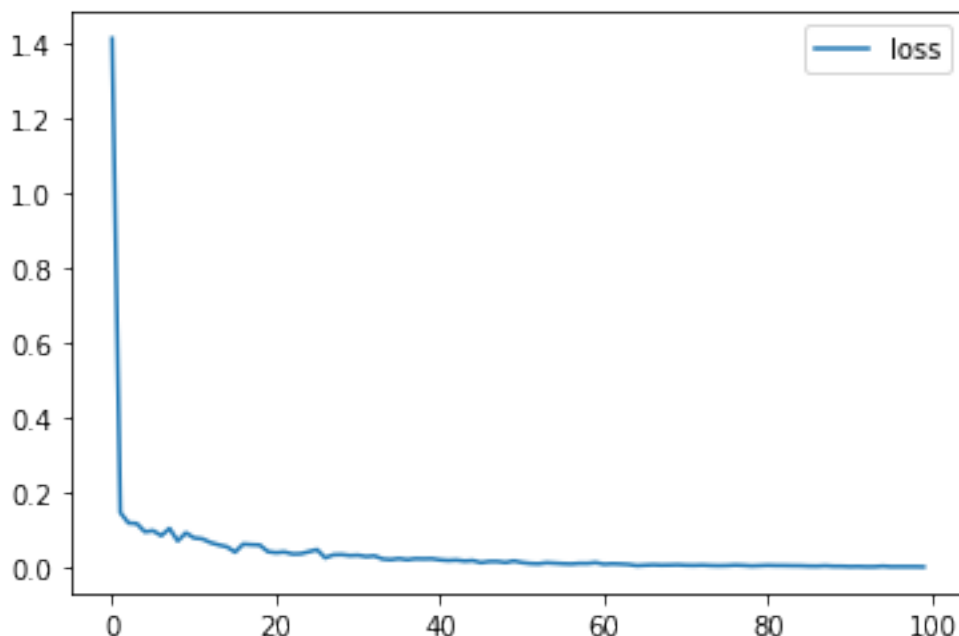
```

```
[24]: train_session.train_losses_[-5:]
```

```
[24]: [0.0036812867, 0.0038135047, 0.0037041684, 0.0037206002, 0.0032002896]
```

```
[25]: import pandas
```

```
pandas.DataFrame({'loss': train_session.train_losses_}).plot();
```



1.6 Forward backward: TrainingAgent

This second implementation uses [TrainingAgent](#).

```
[26]: from onnxcustom.training.optimizers_partial import OrtGradientForwardBackwardOptimizer

train_session = OrtGradientForwardBackwardOptimizer(
    onx, ['init', 'init_1', 'init_b10'], learning_rate=1e-1,
    batch_size=2, max_iter=100)
```

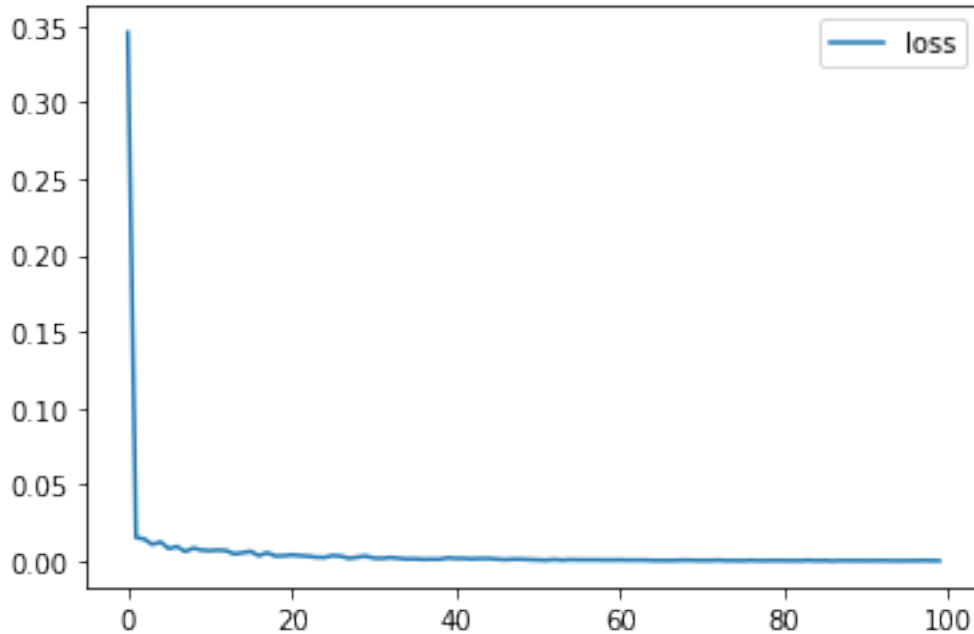
```
[27]: train_session.fit(X, y)
```

```
[27]: OrtGradientForwardBackwardOptimizer(model_onnx='ir_version...',
weights_to_train=['init', 'init_1', 'init_b10'], loss_output_name='loss',
max_iter=100, training_optimizer_name='SGD0Optimizer', batch_size=2,
learning_rate=LearningRateSGD(eta0=0.1, alpha=0.0001, power_t=0.25,
learning_rate='invscaling'), value=0.03162277660168379, device='cpu',
warm_start=False, verbose=0, validation_every=10,
learning_loss=SquareLearningLoss(), enable_logging=False, weight_name=None,
learning_penalty=NoLearningPenalty(), exc=True)
```

```
[28]: train_session.train_losses_[-5:]
```

```
[28]: [0.00040441833, 0.00037421435, 0.00049950054, 0.00042527347, 0.00031072882]
```

```
[29]: pandas.DataFrame({'loss': train_session.train_losses_}).plot();
```



```
[30]: train_session.trained_coef_
```

```
[30]: {'init': <onnxruntime.capi.onnxruntime_pybind11_state.OrtValue at
0x162a9199fb0>,
      'init_1': <onnxruntime.capi.onnxruntime_pybind11_state.OrtValue at
0x162a91a20f0>,
      'init_b10': <onnxruntime.capi.onnxruntime_pybind11_state.OrtValue at
0x162a91a2030>}
```

```
[31]: {k: v.numpy() for k, v in train_session.trained_coef_.items()}
```

```
[31]: {'init': array([[ -0.35357383,  0.6850407 ]], dtype=float32),
      'init_1': array([[ -1.916494 ,  2.8799832 ]], dtype=float32),
      'init_b10': array([ -1.0036615], dtype=float32)}
```

Not the same weights? What about the prediction?

```
[32]: trained_onx = train_session.get_trained_onnx()
```

```
[33]: print(onnx_simple_text_plot(trained_onx))
```

```
opset: domain='' version=14
input: name='x' type=dtype('float32') shape=(0, 2)
init: name='init' type=dtype('float32') shape=(0,) -- array([ -0.35357383,
0.6850407 ], dtype=float32)
init: name='init_1' type=dtype('float32') shape=(0,) -- array([ -1.916494 ,
2.8799832], dtype=float32)
init: name='init_2' type=dtype('int64') shape=(0,) -- array([1], dtype=int64)
init: name='init_3' type=dtype('int64') shape=(0,) -- array([2], dtype=int64)
init: name='init_5' type=dtype('int64') shape=(0,) -- array([0], dtype=int64)
```

```

init: name='init_b10' type=dtype('float32') shape=(0,) -- array([-1.0036615],
dtype=float32)
init: name='init_b11' type=dtype('int64') shape=(0,) -- array([-1,  1],
dtype=int64)
Mul(x, x) -> out_mul_0
  Mul(out_mul_0, init) -> out_mul_0_1
Mul(x, init_1) -> out_mul_0_2
  Add(out_mul_0_2, out_mul_0_1) -> out_add_0
    Slice(out_add_0, init_2, init_3, init_2) -> out_sli_0
      Squeeze(out_sli_0, init_2) -> out_squ_0
    Slice(out_add_0, init_5, init_2, init_2) -> out_sli_0_1
      Squeeze(out_sli_0_1, init_2) -> out_squ_0_1
        Add(out_squ_0_1, out_squ_0) -> out_add_0_1
          Add(out_add_0_1, init_b10) -> out_add_0_2
            Reshape(out_add_0_2, init_b11) -> y
output: name='y' type=dtype('float32') shape=(0, 1)

```

```

[34]: oinf = OnnxInference(trained_onx)
      oinf.run({'x': X})['y'][:5]

```

```

[34]: array([[ -0.6123954],
             [-1.303561 ],
             [-2.0257921],
             [-1.2778704],
             [-0.9708453]], dtype=float32)

```

```

[35]: y[:5]

```

```

[35]: array([[ -0.58675164],
             [-1.3148587 ],
             [-2.0666485 ],
             [-1.272753  ],
             [-0.95404863]], dtype=float32)

```

It works.

1.7 MLPRegressor

```

[36]: import warnings
      import time
      import numpy
      import matplotlib.pyplot as plt
      from pandas import DataFrame
      from onnxruntime import get_device
      from sklearn.datasets import make_regression
      from sklearn.model_selection import train_test_split
      from sklearn.neural_network import MLPRegressor
      from skl2onnx import to_onnx

      X, y = make_regression(1000, n_features=100, bias=2)
      X = X.astype(numpy.float32)
      y = y.astype(numpy.float32)
      X_train, X_test, y_train, y_test = train_test_split(X, y)

```

```
[37]: batch_size = 15
max_iter = 100

nn = MLPRegressor(hidden_layer_sizes=(50, 10), max_iter=max_iter,
                  solver='sgd', learning_rate_init=5e-5,
                  n_iter_no_change=max_iter * 3, batch_size=batch_size,
                  learning_rate="invscaling",
                  # default values
                  momentum=0.9, nesterovs_momentum=True, power_t=0.5)

with warnings.catch_warnings():
    warnings.simplefilter('ignore')
    nn.fit(X_train, y_train)
```

Conversion to ONNX

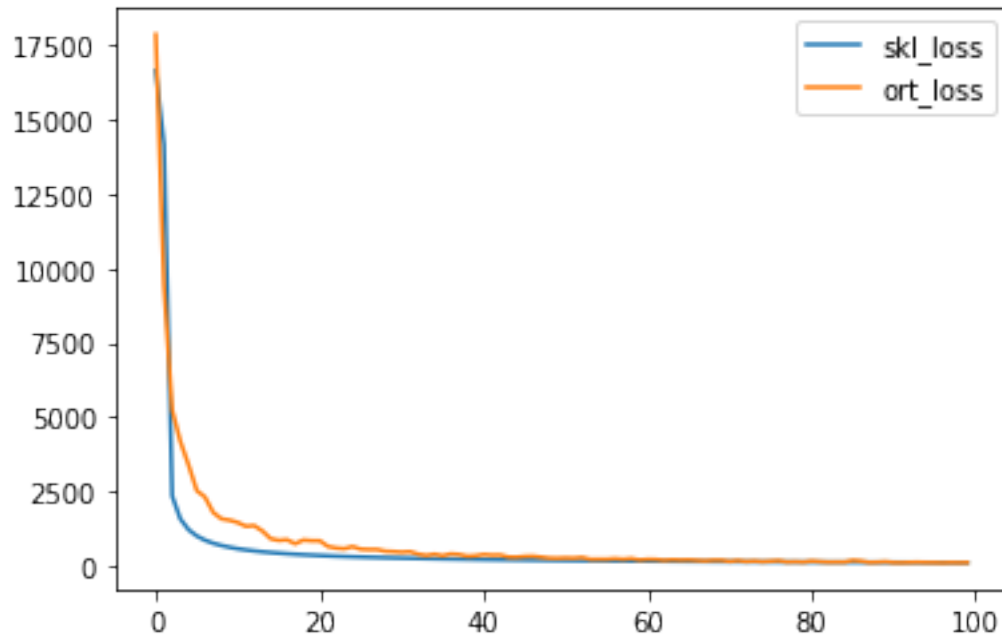
```
[38]: from onnxcustom.utils.onnx_helper import onnx_rename_weights
onx = to_onnx(nn, X_train[:1].astype(numpy.float32), target_opset=15)
onx = onnx_rename_weights(onx)
```

```
[39]: train_session = OrtGradientForwardBackwardOptimizer(
    onx, device='cpu', learning_rate=5e-5,
    warm_start=False, max_iter=max_iter, batch_size=batch_size)
```

```
[40]: train_session.fit(X_train, y_train)
```

```
[40]: OrtGradientForwardBackwardOptimizer(model_onnx='ir_version...',
weights_to_train="['IO_coeff...]", loss_output_name='loss', max_iter=100,
training_optimizer_name='SGDOptimizer', batch_size=15,
learning_rate=LearningRateSGD(eta0=5e-05, alpha=0.0001, power_t=0.25,
learning_rate='invscaling'), value=1.5811388300841898e-05, device='cpu',
warm_start=False, verbose=0, validation_every=10,
learning_loss=SquareLearningLoss(), enable_logging=False, weight_name=None,
learning_penalty=NoLearningPenalty(), exc=True)
```

```
[41]: pandas.DataFrame(dict(skl_loss=nn.loss_curve_, ort_loss=train_session.train_losses_)).
    .plot();
```



```
[42]: %timeit -n 1 -r 1 nn.fit(X_train, y_train)
```

```
C:\Python395_x64\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:692:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

1.98 s ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
```

```
[43]: %timeit -n 1 -r 1 train_session.fit(X_train, y_train)
```

```
1.88 s ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
```

1.8 Not exactly the same: Nesterov?

```
[44]: from onnxcustom.training.sgd_learning_rate import LearningRateSGDNesterov

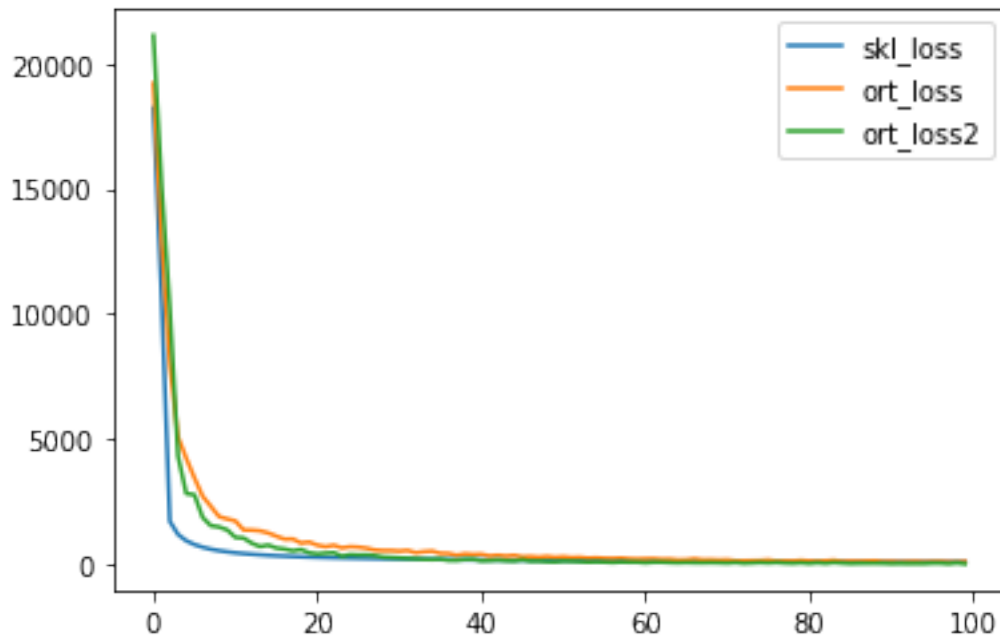
train_session2 = OrtGradientForwardBackwardOptimizer(
    onnx, device='cpu', warm_start=False, max_iter=max_iter, batch_size=batch_size,
    learning_rate=LearningRateSGDNesterov(1e-5, nesterov=True, momentum=0.9))
```

```
[45]: train_session2.fit(X_train, y_train)
```

```
[45]: OrtGradientForwardBackwardOptimizer(model_onnx='ir_version...',
weights_to_train="['IO_coeff...'", loss_output_name='loss', max_iter=100,
training_optimizer_name='SGDOptimizer', batch_size=15,
learning_rate=LearningRateSGDNesterov(eta0=1e-05, alpha=0.0001, power_t=0.25,
```

```
learning_rate='invscaling', momentum=0.9, nesterov=True),
value=3.162277660168379e-06, device='cpu', warm_start=False, verbose=0,
validation_every=10, learning_loss=SquareLearningLoss(), enable_logging=False,
weight_name=None, learning_penalty=NoLearningPenalty(), exc=True)
```

```
[46]: pandas.DataFrame(dict(skl_loss=nn.loss_curve_,
                           ort_loss=train_session.train_losses_,
                           ort_loss2=train_session2.train_losses_)).plot();
```



```
[47]: %timeit -n 1 -r 1 train_session2.fit(X_train, y_train)
```

2.26 s ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)

1.9 Profiling

```
[48]: def clean_name(text):
    pos = text.find('onnxruntime')
    if pos >= 0:
        return text[pos:]
    pos = text.find('sklearn')
    if pos >= 0:
        return text[pos:]
    pos = text.find('onnxcustom')
    if pos >= 0:
        return text[pos:]
    pos = text.find('site-packages')
    if pos >= 0:
        return text[pos:]
    return text
```



```
from pyquickhelper.pycode.profilng import profile, profile2graph
```

```
ps = profile(lambda:train_session2.fit(X, y))[0]
root, nodes = profile2graph(ps, clean_text=clean_name)
text = root.to_text()
print(text)
```

```
<lambda> -- 1 1
-- 0.00001 3.78074 -- <ipython-input-81-1255a3a5f723>:18:<lambda> (<lambda>)
    fit -- 1 1
-- 0.00181 3.78073 --
onnxcustom/onnxcustom/training/optimizers_partial.py:263:fit (fit)
    __init__ -- 1 1
-- 0.00002 0.00003 -- onnxcustom/onnxcustom/training/data_loader.py:26:__init__
(__init__)
    get_ort_device -- 1 1
-- 0.00000 0.00000 -- onnxruntime_helper.py:55:get_ort_device (get_ort_device)
    numpy_to_ort_value -- 2 2
-- 0.00000 0.00001 -- onnxruntime_helper.py:120:numpy_to_ort_value
(numpy_to_ort_value) +++
    needs_grad -- 3 3
-- 0.00001 0.00001 --
onnxcustom/onnxcustom/training/optimizers_partial.py:99:needs_grad (needs_grad)
    needs_grad -- 3 3
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:299:needs_grad (needs_grad)
    get_full_state -- 101 101
-- 0.00020 0.00093 --
onnxcustom/onnxcustom/training/optimizers_partial.py:147:get_full_state
(get_full_state) +++
    set_state -- 4 4
-- 0.00008 0.00026 --
onnxcustom/onnxcustom/training/optimizers_partial.py:196:set_state (set_state)
    _get_att_state -- 4 4
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/optimizers_partial.py:139:_get_att_state
(_get_att_state) +++
    numpy_to_ort_value -- 24 24
-- 0.00002 0.00011 -- onnxruntime_helper.py:120:numpy_to_ort_value
(numpy_to_ort_value) +++
    <built-in method numpy.zeros> -- 12 12
-- 0.00002 0.00002 -- ~:0:<built-in method numpy.zeros> (<built-in method
numpy.zeros>)
    <method 'append' of 'list' objects> -- 56 56
-- 0.00001 0.00001 -- ~:0:<method 'append' of 'list' objects> (<method 'append'
of 'list' objects>) +++
    <built-in method builtins.isinstance> -- 24 24
-- 0.00000 0.00000 -- ~:0:<built-in method builtins.isinstance> (<built-in
method builtins.isinstance>) +++
    <listcomp> -- 1 1
-- 0.00001 0.00095 --
onnxcustom/onnxcustom/training/optimizers_partial.py:311:<listcomp> (<listcomp>)
    get_initializer -- 7 7
```

```

-- 0.00004 0.00094 --
onnxcustom/onnxcustom/training/ortgradient.py:269:get_initializer
(get_initializer) +++
    <listcomp> -- 1 1
-- 0.00001 0.00083 --
onnxcustom/onnxcustom/training/optimizers_partial.py:315:<listcomp> (<listcomp>)
    get_initializer -- 7 7
-- 0.00004 0.00082 --
onnxcustom/onnxcustom/training/ortgradient.py:269:get_initializer
(get_initializer) +++
    _iteration -- 100 100
-- 0.41903 3.74610 --
onnxcustom/onnxcustom/training/optimizers_partial.py:397:_iteration (_iteration)
    iter_ortvalue -- 6800 6800
-- 0.02838 0.14761 --
onnxcustom/onnxcustom/training/data_loader.py:139:iter_ortvalue (iter_ortvalue)
    _next_iter -- 6700 6700
-- 0.00946 0.07207 --
onnxcustom/onnxcustom/training/data_loader.py:93:_next_iter (_next_iter)
    <built-in method builtins.len> -- 6700 6700
-- 0.00245 0.00423 -- ~:0:<built-in method builtins.len> (<built-in method
builtins.len>) +++
    <method 'randint' o...domState' objects> -- 6700 6700
-- 0.05838 0.05838 -- ~:0:<method 'randint' of 'numpy.random.mtrand.RandomState'
objects> (<method 'randint' of 'numpy.random.mtrand.RandomState' objects>)
    numpy_to_ort_value -- 13400 13400
-- 0.00658 0.03860 -- onnxruntime_helper.py:120:numpy_to_ort_value
(numpy_to_ort_value) +++
    <built-in method builtins.len> -- 6900 6900
-- 0.00467 0.00855 -- ~:0:<built-in method builtins.len> (<built-in method
builtins.len>) +++
    forward -- 6700 6700
-- 0.31685 0.44643 -- onnxcustom/onnxcustom/training/ortgradient.py:623:forward
(forward)
    input_to_ort -- 6700 6700
-- 0.08002 0.11492 --
onnxcustom/onnxcustom/training/ortgradient.py:552:input_to_ort (input_to_ort)
+++
    save_for_backward -- 6700 6700
-- 0.01032 0.01032 --
onnxcustom/onnxcustom/training/ortgradient.py:604:save_for_backward
(save_for_backward)
    <method 'append' of 'list' objects> -- 6700 6700
-- 0.00434 0.00434 -- ~:0:<method 'append' of 'list' objects> (<method 'append'
of 'list' objects>) +++
    backward -- 6700 6700
-- 0.43012 0.48957 -- onnxcustom/onnxcustom/training/ortgradient.py:702:backward
(backward)
    input_to_ort -- 6700 6700
-- 0.04148 0.05262 --
onnxcustom/onnxcustom/training/ortgradient.py:552:input_to_ort (input_to_ort)
+++
    saved_tensors -- 6700 6700
-- 0.00207 0.00207 --

```

```

onnxcustom/onnxcustom/training/ortgradient.py:613:saved_tensors (saved_tensors)
    <method 'pop' of 'list' objects> -- 6700 6700
-- 0.00476 0.00476 -- ~:0:<method 'pop' of 'list' objects> (<method 'pop' of
'list' objects>)
    loss_gradient -- 6700 6700
-- 0.05841 0.26967 --
onnxcustom/onnxcustom/training/sgd_learning_loss.py:53:loss_gradient
(loss_gradient)
    clear_binding_inputs -- 6700 6700
-- 0.00545 0.01270 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:130:clear_binding_inputs
(clear_binding_inputs)
    _cache_in_clear -- 6700 6700
-- 0.00568 0.00725 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:119:_cache_in_clear
(_cache_in_clear)
    <built-in method builtins.id> -- 6700 6700
-- 0.00157 0.00157 -- ~:0:<built-in method builtins.id> (<built-in method
builtins.id>) +++
    _bind_input_ortvalue -- 13400 13400
-- 0.02070 0.07545 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:159:_bind_input_ortvalue
(_bind_input_ortvalue) +++
    _call_iobinding -- 6700 6700
-- 0.11997 0.11997 --
onnxcustom/onnxcustom/training/sgd_learning_loss.py:50:_call_iobinding
(_call_iobinding)
    <built-in method builtins.hasattr> -- 13400 13400
-- 0.00315 0.00315 -- ~:0:<built-in method builtins.hasattr> (<built-in method
builtins.hasattr>) +++
    penalty_loss -- 6700 6700
-- 0.00112 0.00112 --
onnxcustom/onnxcustom/training/sgd_learning_penalty.py:84:penalty_loss
(penalty_loss)
    update_weights -- 40200 40200
-- 0.00651 0.00651 --
onnxcustom/onnxcustom/training/sgd_learning_penalty.py:95:update_weights
(update_weights)
    update_weights -- 40200 40200
-- 0.40487 1.94238 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:345:update_weights
(update_weights)
    _bind_input_ortvalue -- 201000 201000
-- 0.19630 0.51693 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:159:_bind_input_ortvalue
(_bind_input_ortvalue) +++
    _bind_output_ortvalue -- 80400 80400
-- 0.07458 0.18952 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:202:_bind_output_ortvalue
(_bind_output_ortvalue)
    _bio_cache -- 80400 80400
-- 0.04417 0.05406 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:138:_bio_cache
(_bio_cache) +++

```

```

        _bio_ptr -- 80400 80400
-- 0.05222 0.05222 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:155:_bio_ptr (_bio_ptr)
+++
        _bio_do_bind_out -- 12 12
-- 0.00003 0.00003 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:198:_bio_do_bind_out
(_bio_do_bind_out)
        <built-in method builtins.isinstance> -- 80400 80400
-- 0.00863 0.00863 -- ~:0:<built-in method builtins.isinstance> (<built-in
method builtins.isinstance>) +++
        _call_iobinding -- 40200 40200
-- 0.63987 0.63987 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:28:_call_iobinding
(_call_iobinding)
        value -- 40200 40200
-- 0.00953 0.00953 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:160:value (value) +++
        <built-in method onnx..ortvalue_from_numpy> -- 80400 80400
-- 0.16512 0.16512 -- ~:0:<built-in method
onnxruntime.capi.onnxruntime_pybind11_state.ortvalue_from_numpy> (<built-in
method onnxruntime.capi.onnxruntime_pybind11_state.ortvalue_from_numpy>) +++
        <built-in method builtins.hasattr> -- 80400 80400
-- 0.01655 0.01655 -- ~:0:<built-in method builtins.hasattr> (<built-in method
builtins.hasattr>) +++
        <method 'mean' of 'numpy.ndarray' objects> -- 100 100
-- 0.00026 0.00426 -- ~:0:<method 'mean' of 'numpy.ndarray' objects> (<method
'mean' of 'numpy.ndarray' objects>)
        _mean -- 100 100
-- 0.00163 0.00400 -- site-packages/numpy/core/_methods.py:162:_mean (_mean)
        _count_reduce_items -- 100 100
-- 0.00097 0.00107 -- site-
packages/numpy/core/_methods.py:66:_count_reduce_items (_count_reduce_items)
        <built-in method ..lize_axis_index> -- 200 200
-- 0.00010 0.00010 -- ~:0:<built-in method
numpy.core._multiarray_umath.normalize_axis_index> (<built-in method
numpy.core._multiarray_umath.normalize_axis_index>)
        <built-in method numpy.asanyarray> -- 100 100
-- 0.00004 0.00004 -- ~:0:<built-in method numpy.asanyarray> (<built-in method
numpy.asanyarray>)
        <method 'reduce' of..py.ufunc' objects> -- 100 100
-- 0.00109 0.00109 -- ~:0:<method 'reduce' of 'numpy.ufunc' objects> (<method
'reduce' of 'numpy.ufunc' objects>)
        <built-in method builtins.hasattr> -- 100 100
-- 0.00006 0.00006 -- ~:0:<built-in method builtins.hasattr> (<built-in method
builtins.hasattr>) +++
        <built-in method builtins.isinstance> -- 100 100
-- 0.00004 0.00004 -- ~:0:<built-in method builtins.isinstance> (<built-in
method builtins.isinstance>) +++
        <built-in method builtins.issubclass> -- 200 200
-- 0.00007 0.00007 -- ~:0:<built-in method builtins.issubclass> (<built-in
method builtins.issubclass>)
        <built-in method numpy.array> -- 100 100
-- 0.00358 0.00358 -- ~:0:<built-in method numpy.array> (<built-in method

```

```

numpy.array>)
    <method 'append' of 'list' objects>                --      6700      6700
-- 0.00169 0.00169 -- ~:0:<method 'append' of 'list' objects> (<method 'append'
of 'list' objects>) +++
    <built-in method builtins.len>                    --      40300      40300
-- 0.01424 0.01424 -- ~:0:<built-in method builtins.len> (<built-in method
builtins.len>) +++
    _create_training_session                          --          1          1
-- 0.00001 0.02824 -- onnxcustom/onnxcustom/training/optimizers_partial.py:626:_
create_training_session (_create_training_session)
    __init__                                           --          1          1
-- 0.00008 0.02820 -- onnxcustom/onnxcustom/training/ortgradient.py:54:__init__
(__init__)
    <listcomp>                                         --          1          1
-- 0.00001 0.00001 --
onnxcustom/onnxcustom/training/ortgradient.py:91:<listcomp> (<listcomp>)
    <listcomp>                                         --          1          1
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/ortgradient.py:94:<listcomp> (<listcomp>)
    <listcomp>                                         --          1          1
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/ortgradient.py:113:<listcomp> (<listcomp>)
    _init_next                                         --          1          1
-- 0.00010 0.02809 --
onnxcustom/onnxcustom/training/ortgradient.py:163:_init_next (_init_next)
    <listcomp>                                         --          1          1
-- 0.00001 0.00001 --
onnxcustom/onnxcustom/training/ortgradient.py:173:<listcomp> (<listcomp>)
    <listcomp>                                         --          1          1
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/ortgradient.py:175:<listcomp> (<listcomp>)
    <listcomp>                                         --          1          1
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/ortgradient.py:178:<listcomp> (<listcomp>)
    _create_onnx_graphs                               --          1          1
-- 0.00662 0.02797 --
onnxcustom/onnxcustom/training/ortgradient.py:287:_create_onnx_graphs
(_create_onnx_graphs)
    <listcomp>                                         --          1          1
-- 0.00001 0.00001 --
onnxcustom/onnxcustom/training/ortgradient.py:396:<listcomp> (<listcomp>)
    <listcomp>                                         --          1          1
-- 0.00001 0.00001 --
onnxcustom/onnxcustom/training/ortgradient.py:397:<listcomp> (<listcomp>)
    <listcomp>                                         --          1          1
-- 0.00001 0.00002 --
onnxcustom/onnxcustom/training/ortgradient.py:399:<listcomp> (<listcomp>)
    _provider_name_to_device_type                    --          1          1
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/ortgradient.py:260:_provider_name_to_device_type
(_provider_name_to_device_type) +++
    <listcomp>                                         --          1          1
-- 0.00002 0.00002 --
onnxcustom/onnxcustom/training/ortgradient.py:404:<listcomp> (<listcomp>)

```

```

-- 0.00000 0.00000 --      _provider_name_to_device_type      --      7      7
onnxcustom/onnxcustom/training/ortgradient.py:260:_provider_name_to_device_type
(_provider_name_to_device_type) +++
      <listcomp>      --      1      1
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/ortgradient.py:410:<listcomp> (<listcomp>)
      _provider_name_to_device_type      --      1      1
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/ortgradient.py:260:_provider_name_to_device_type
(_provider_name_to_device_type) +++
      <listcomp>      --      1      1
-- 0.00001 0.00001 --
onnxcustom/onnxcustom/training/ortgradient.py:479:<listcomp> (<listcomp>)
      <listcomp>      --      1      1
-- 0.00001 0.00001 --
onnxcustom/onnxcustom/training/ortgradient.py:480:<listcomp> (<listcomp>)
      get_inputs      --      1      1
-- 0.00000 0.00000 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi
/onnxruntime_inference_collection.py:111:get_inputs (get_inputs)
      get_outputs      --      1      1
-- 0.00000 0.00000 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi
/onnxruntime_inference_collection.py:115:get_outputs (get_outputs)
      __init__      --      2      2
-- 0.00004 0.02063 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi
/onnxruntime_inference_collection.py:283:__init__ (__init__)
      get      --      2      2
-- 0.00001 0.00004 -- C:/Python395_x64/lib/_collections_abc.py:759:get (get)
      __getitem__      --      2      2
-- 0.00001 0.00003 -- C:/Python395_x64/lib/os.py:674:__getitem__ (__getitem__)
      encodekey      --      2      2
-- 0.00001 0.00002 -- C:/Python395_x64/lib/os.py:746:encodekey (encodekey)
      check_str      --      2      2
-- 0.00000 0.00000 -- C:/Python395_x64/lib/os.py:740:check_str (check_str)
      __init__      --      2      2
-- 0.00000 0.00000 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi
/onnxruntime_inference_collection.py:101:__init__ (__init__)
      _create_inference_session      --      2      2
-- 0.02045 0.02055 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi
/onnxruntime_inference_collection.py:346:_create_inference_session
(_create_inference_session)
      check_and_normalize_provider_args      --      2      2
-- 0.00004 0.00008 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi
/onnxruntime_inference_collection.py:25:check_and_normalize_provider_args
(check_and_normalize_provider_args)
      set_provider_options      --      2      2
-- 0.00001 0.00001 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi
/onnxruntime_inference_collection.py:53:set_provider_options
(set_provider_options)
      <dictcomp>      --      2      2
-- 0.00000 0.00000 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi
/onnxruntime_inference_collection.py:62:<dictcomp> (<dictcomp>)
      <listcomp>      --      2      2
-- 0.00000 0.00000 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi

```

```

/onnxruntime_inference_collection.py:75:<listcomp> (<listcomp>)
      <listcomp> -- 2 2
-- 0.00000 0.00000 -- onnxruntime/build/Windows/Release/Release/onnxruntime/capi
/onnxruntime_inference_collection.py:78:<listcomp> (<listcomp>)
      load_model -- 2 2
-- 0.00001 0.00049 -- site-packages/onnx/__init__.py:107:load_model (load_model)
      _load_bytes -- 2 2
-- 0.00002 0.00003 -- site-packages/onnx/__init__.py:30:_load_bytes
(_load_bytes)
      inner -- 4 4
-- 0.00000 0.00000 -- C:/Python395_x64/lib/typing.py:262:inner (inner) +++
      cast -- 4 4
-- 0.00000 0.00000 -- C:/Python395_x64/lib/typing.py:1333:cast (cast) +++
      _get_file_path -- 2 2
-- 0.00000 0.00000 -- site-packages/onnx/__init__.py:50:_get_file_path
(_get_file_path)
      load_model_from_string -- 2 2
-- 0.00001 0.00045 -- site-packages/onnx/__init__.py:147:load_model_from_string
(load_model_from_string)
      _deserialize -- 2 2
-- 0.00001 0.00044 -- site-packages/onnx/__init__.py:81:_deserialize
(_deserialize)
      inner -- 2 2
-- 0.00000 0.00000 -- C:/Python395_x64/lib/typing.py:262:inner (inner) +++
      cast -- 2 2
-- 0.00000 0.00000 -- C:/Python395_x64/lib/typing.py:1333:cast (cast) +++
      <method 'Pa...' objects> -- 2 2
-- 0.00042 0.00042 -- ~:0:<method 'ParseFromString' of
'google.protobuf.pyext._message.CMessage' objects> (<method 'ParseFromString' of
'google.protobuf.pyext._message.CMessage' objects>)
      <built-in method builtins.len> -- 16 16
-- 0.00000 0.00000 -- ~:0:<built-in method builtins.len> (<built-in method
builtins.len>) +++
      <method 'Serializ...essage' objects> -- 1 1
-- 0.00014 0.00014 -- ~:0:<method 'SerializeToString' of
'google.protobuf.pyext._message.CMessage' objects> (<method 'SerializeToString'
of 'google.protobuf.pyext._message.CMessage' objects>)
      new_instance -- 1 1
-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/ortgradient.py:211:new_instance (new_instance)
      __init__ -- 1 1
-- 0.00000 0.00000 -- onnxcustom/onnxcustom/training/ortgradient.py:501:__init__
(__init__)
      device_to_providers -- 1 1
-- 0.00003 0.00003 -- onnxruntime_helper.py:133:device_to_providers
(device_to_providers)
      value -- 100 100
-- 0.00003 0.00003 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:160:value (value) +++
      init_learning_rate -- 1 1
-- 0.00000 0.00001 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:307:init_learning_rate
(init_learning_rate)
      init_learning_rate -- 1 1

```

```

-- 0.00000 0.00000 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:176:init_learning_rate
(init_learning_rate)
    update_learning_rate                                --      100      100
-- 0.00015 0.00098 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:314:update_learning_rate
(update_learning_rate)
    update_learning_rate                                --      100      100
-- 0.00084 0.00084 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:194:update_learning_rate
(update_learning_rate)
    proto_type_to_dtype                                --         6         6
-- 0.00001 0.00001 --
onnxcustom/onnxcustom/utils/onnx_helper.py:53:proto_type_to_dtype
(proto_type_to_dtype)
    <method 'append' of 'list' objects>                  --      107      107
-- 0.00003 0.00003 -- ~:0:<method 'append' of 'list' objects> (<method 'append'
of 'list' objects>) +++
    <built-in method builtins.len>                      --      108      108
-- 0.00002 0.00002 -- ~:0:<built-in method builtins.len> (<built-in method
builtins.len>) +++
    <method 'randn' of 'numpy.nd.RandomState' objects> --         6         6
-- 0.00040 0.00040 -- ~:0:<method 'randn' of 'numpy.random.mtrand.RandomState'
objects> (<method 'randn' of 'numpy.random.mtrand.RandomState' objects>)
inner                                                    --         6         6
-- 0.00001 0.00001 -- C:/Python395_x64/lib/typing.py:262:inner (inner)
cast                                                    --         6         6
-- 0.00000 0.00000 -- C:/Python395_x64/lib/typing.py:1333:cast (cast)
_bio_cache                                              --    294800    294800
-- 0.18126 0.22052 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:138:_bio_cache
(_bio_cache)
    <built-in method builtins.id>                        --    294800    294800
-- 0.03926 0.03926 -- ~:0:<built-in method builtins.id> (<built-in method
builtins.id>) +++
_bio_ptr                                              --    294800    294800
-- 0.20762 0.20762 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:155:_bio_ptr (_bio_ptr)
_bind_input_ortvalue                                    --    214400    214400
-- 0.21699 0.59239 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:159:_bind_input_ortvalue
(_bind_input_ortvalue)
    _bio_cache                                          --    214400    214400
-- 0.13709 0.16646 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:138:_bio_cache
(_bio_cache) +++
    _bio_do_bind_in                                    --     14000     14000
-- 0.03012 0.03012 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:151:_bio_do_bind_in
(_bio_do_bind_in)
    _bio_ptr                                          --    214400    214400
-- 0.15540 0.15540 --
onnxcustom/onnxcustom/training/_base_onnx_function.py:155:_bio_ptr (_bio_ptr)
+++

```



```

    <built-in method builtins.isinstance> -- 214400 214400
-- 0.02341 0.02341 -- ~:0:<built-in method builtins.isinstance> (<built-in
method builtins.isinstance>) +++
_get_att_state -- 205 205
-- 0.00007 0.00007 --
onnxcustom/onnxcustom/training/optimizers_partial.py:139:_get_att_state
(_get_att_state)
get_full_state -- 101 301
-- 0.00049 0.00093 --
onnxcustom/onnxcustom/training/optimizers_partial.py:147:get_full_state
(get_full_state)
_get_att_state -- 201 201
-- 0.00007 0.00007 --
onnxcustom/onnxcustom/training/optimizers_partial.py:139:_get_att_state
(_get_att_state) +++
    <listcomp> -- 100 100
-- 0.00021 0.00072 --
onnxcustom/onnxcustom/training/optimizers_partial.py:152:<listcomp> (<listcomp>)
    get_full_state -- 200 200
-- 0.00030 0.00050 --
onnxcustom/onnxcustom/training/optimizers_partial.py:147:get_full_state
(get_full_state) +++
    <built-in method builtins.getattr> -- 201 201
-- 0.00004 0.00004 -- ~:0:<built-in method builtins.getattr> (<built-in method
builtins.getattr>) +++
    <built-in method builtins.hasattr> -- 201 201
-- 0.00005 0.00005 -- ~:0:<built-in method builtins.hasattr> (<built-in method
builtins.hasattr>) +++
    <built-in method builtins.isinstance> -- 301 301
-- 0.00007 0.00007 -- ~:0:<built-in method builtins.isinstance> (<built-in
method builtins.isinstance>) +++
_provider_name_to_device_type -- 9 9
-- 0.00001 0.00001 --
onnxcustom/onnxcustom/training/ortgradient.py:260:_provider_name_to_device_type
(_provider_name_to_device_type)
get_initializer -- 14 14
-- 0.00008 0.00175 --
onnxcustom/onnxcustom/training/ortgradient.py:269:get_initializer
(get_initializer)
    to_array -- 12 12
-- 0.00009 0.00168 -- site-packages/onnx/numpy_helper.py:21:to_array (to_array)
    uses_external_data -- 12 12
-- 0.00001 0.00001 -- site-
packages/onnx/external_data_helper.py:224:uses_external_data
(uses_external_data)
    <method 'HasField' of '...age.CMessage' objects> -- 12 12
-- 0.00000 0.00000 -- ~:0:<method 'HasField' of
'google.protobuf.pyext._message.CMessage' objects> (<method 'HasField' of
'google.protobuf.pyext._message.CMessage' objects>) +++
    <method 'astype' of 'numpy.ndarray' objects> -- 12 12
-- 0.00006 0.00006 -- ~:0:<method 'astype' of 'numpy.ndarray' objects> (<method
'astype' of 'numpy.ndarray' objects>) +++
    <method 'reshape' of 'numpy.ndarray' objects> -- 12 12
-- 0.00002 0.00002 -- ~:0:<method 'reshape' of 'numpy.ndarray' objects> (<method

```

```

'reshape' of 'numpy.ndarray' objects>) +++
    <built-in method numpy.asarray> -- 12 12
-- 0.00148 0.00148 -- ~:0:<built-in method numpy.asarray> (<built-in method
numpy.asarray>)
    <built-in method builtins.getattr> -- 12 12
-- 0.00001 0.00001 -- ~:0:<built-in method builtins.getattr> (<built-in method
builtins.getattr>) +++
    <method 'HasField' of 'google.protobuf.pyext._message.CMessage' objects> -- 24 24
-- 0.00001 0.00001 -- ~:0:<method 'HasField' of
'google.protobuf.pyext._message.CMessage' objects> (<method 'HasField' of
'google.protobuf.pyext._message.CMessage' objects>) +++
input_to_ort -- 13400 13400
-- 0.12150 0.16754 --
onnxcustom/onnxcustom/training/ortgradient.py:552:input_to_ort (input_to_ort)
    <built-in method builtins.all> -- 13400 13400
-- 0.01681 0.03690 -- ~:0:<built-in method builtins.all> (<built-in method
builtins.all>) +++
    <built-in method builtins.isinstance> -- 13400 13400
-- 0.00712 0.00712 -- ~:0:<built-in method builtins.isinstance> (<built-in
method builtins.isinstance>) +++
    <built-in method builtins.len> -- 13400 13400
-- 0.00202 0.00202 -- ~:0:<built-in method builtins.len> (<built-in method
builtins.len>) +++
value -- 40300 40300
-- 0.00955 0.00955 --
onnxcustom/onnxcustom/training/sgd_learning_rate.py:160:value (value)
numpy_to_ort_value -- 13426 13426
-- 0.00661 0.03872 -- onnxruntime_helper.py:120:numpy_to_ort_value
(numpy_to_ort_value)
    <built-in method onnxruntime...state.ortvalue_from_numpy> -- 13426 13426
-- 0.03211 0.03211 -- ~:0:<built-in method
onnxruntime.capi.onnxruntime_pybind11_state.ortvalue_from_numpy> (<built-in
method onnxruntime.capi.onnxruntime_pybind11_state.ortvalue_from_numpy>) +++
<method 'astype' of 'numpy.ndarray' objects> -- 18 18
-- 0.00014 0.00014 -- ~:0:<method 'astype' of 'numpy.ndarray' objects> (<method
'astype' of 'numpy.ndarray' objects>)
<method 'append' of 'list' objects> -- 13575 13575
-- 0.00608 0.00608 -- ~:0:<method 'append' of 'list' objects> (<method 'append'
of 'list' objects>)
<built-in method builtins.hasattr> -- 94120 94120
-- 0.01981 0.01981 -- ~:0:<built-in method builtins.hasattr> (<built-in method
builtins.hasattr>)
<built-in method builtins.isinstance> -- 362251 362251
-- 0.04476 0.04477 -- ~:0:<built-in method builtins.isinstance> (<built-in
method builtins.isinstance>)
__instancecheck__ -- 4 4
-- 0.00001 0.00001 -- C:/Python395_x64/lib/abc.py:96:__instancecheck__
(__instancecheck__)
<built-in method builtins.len> -- 67437 67437
-- 0.02341 0.02908 -- ~:0:<built-in method builtins.len> (<built-in method
builtins.len>)
__len__ -- 13600 13600
-- 0.00567 0.00567 -- onnxcustom/onnxcustom/training/data_loader.py:89:__len__
(__len__)

```

```

<method 'reshape' of 'numpy.ndarray' objects> -- 14 14
-- 0.00002 0.00002 -- ~:0:<method 'reshape' of 'numpy.ndarray' objects> (<method
'reshape' of 'numpy.ndarray' objects>)
<built-in method builtins.getattr> -- 213 213
-- 0.00005 0.00005 -- ~:0:<built-in method builtins.getattr> (<built-in method
builtins.getattr>)
<built-in method onnxruntime...1_state.ortvalue_from_numpy> -- 93826 93826
-- 0.19723 0.19723 -- ~:0:<built-in method
onnxruntime.capi.onnxruntime_pybind11_state.ortvalue_from_numpy> (<built-in
method onnxruntime.capi.onnxruntime_pybind11_state.ortvalue_from_numpy>)
<built-in method builtins.id> -- 301501 301501
-- 0.04083 0.04083 -- ~:0:<built-in method builtins.id> (<built-in method
builtins.id>)
<method 'HasField' of 'google...message.CMessage' objects> -- 36 36
-- 0.00001 0.00001 -- ~:0:<method 'HasField' of
'google.protobuf.pyext._message.CMessage' objects> (<method 'HasField' of
'google.protobuf.pyext._message.CMessage' objects>)
<built-in method builtins.all> -- 13404 13404
-- 0.01681 0.03690 -- ~:0:<built-in method builtins.all> (<built-in method
builtins.all>)
    <lambda> -- 53600 53600
-- 0.01461 0.02009 -- onnxcustom/onnxcustom/training/ortgradient.py:572:<lambda>
(<lambda>)
    <built-in method builtins.isinstance> -- 53600 53600
-- 0.00548 0.00548 -- ~:0:<built-in method builtins.isinstance> (<built-in
method builtins.isinstance>) +++

    _iteration -- 100 100 -- 0.41903 3.74610 --
        iter_ortvalue -- 6800 6800 -- 0.02838 0.14761 --
            _next_iter -- 6700 6700 -- 0.00946 0.07207 --
                <built-in method builtins.len> -- 6700 6700 -- 0.00245 0.00423 --
                    <method 'randint' o...domState' objects> -- 6700 6700 -- 0.05838 0.05838 --
                        numpy_to_ort_value -- 13400 13400 -- 0.00658 0.03860 --
                            <built-in method builtins.len> -- 6900 6900 -- 0.00467 0.00855 --
                                forward -- 6700 6700 -- 0.31685 0.44643 --
                                    input_to_ort -- 6700 6700 -- 0.08002 0.11492 --
                                        save_for_backward -- 6700 6700 -- 0.01032 0.01032 --
                                            <method 'append' of 'list' objects> -- 6700 6700 -- 0.00434 0.00434 --
                                                backward -- 6700 6700 -- 0.43012 0.48957 --
                                                    input_to_ort -- 6700 6700 -- 0.04148 0.05262 --
                                                        saved_tensors -- 6700 6700 -- 0.00207 0.00207 --
                                                            <method 'pop' of 'list' objects> -- 6700 6700 -- 0.00476 0.00476 --
                                                                loss_gradient -- 6700 6700 -- 0.05841 0.26967 --
                                                                    clear_binding_inputs -- 6700 6700 -- 0.00545 0.01270 --
                                                                        _cache_in_clear -- 6700 6700 -- 0.00568 0.00725 --
                                                                            <built-in method builtins.id> -- 6700 6700 -- 0.00157 0.00157 --
                                                                                _bind_input_ortvalue -- 13400 13400 -- 0.02070 0.07545 --
                                                                                    _call_iobinding -- 6700 6700 -- 0.11997 0.11997 --
                                                                                        <built-in method builtins.hasattr> -- 13400 13400 -- 0.00315 0.00315 --
                                                                                            penalty_loss -- 6700 6700 -- 0.00112 0.00112 --
                                                                                                update_weights -- 40200 40200 -- 0.00651 0.00651 --
                                                                                                    update_weights -- 40200 40200 -- 0.40487 1.94238 --
                                                                                                        _bind_input_ortvalue -- 201000 201000 -- 0.19630 0.51693 --
                                                                                                            _bind_output_ortvalue -- 80400 80400 -- 0.07458 0.18952 --

```

_bio_cache	--	80400	80400	--	0.04417	0.05406	--
_bio_ptr	--	80400	80400	--	0.05222	0.05222	--
_bio_do_bind_out	--	12	12	--	0.00003	0.00003	--
<built-in method builtins.isinstance>	--	80400	80400	--	0.00863	0.00863	--
_call_iobinding	--	40200	40200	--	0.63987	0.63987	--
value	--	40200	40200	--	0.00953	0.00953	--
<built-in method onnx...ortvalue_from_numpy>	--	80400	80400	--	0.16512	0.16512	--
<built-in method builtins.hasattr>	--	80400	80400	--	0.01655	0.01655	--
<method 'mean' of 'numpy.ndarray' objects>	--	100	100	--	0.00026	0.00426	--
_mean	--	100	100	--	0.00163	0.00400	--
_count_reduce_items	--	100	100	--	0.00097	0.00107	--
<built-in method ...lize_axis_index>	--	200	200	--	0.00010	0.00010	--
<built-in method numpy.asanyarray>	--	100	100	--	0.00004	0.00004	--
<method 'reduce' of ...py.ufunc' objects>	--	100	100	--	0.00109	0.00109	--
<built-in method builtins.hasattr>	--	100	100	--	0.00006	0.00006	--
<built-in method builtins.isinstance>	--	100	100	--	0.00004	0.00004	--
<built-in method builtins.issubclass>	--	200	200	--	0.00007	0.00007	--
<built-in method numpy.array>	--	100	100	--	0.00358	0.00358	--
<method 'append' of 'list' objects>	--	6700	6700	--	0.00169	0.00169	--
<built-in method builtins.len>	--	40300	40300	--	0.01424	0.01424	--
_create_training_session	--	1	1	--	0.00001	0.02824	--
__init__	--	1	1	--	0.00008	0.02820	--
<listcomp>	--	1	1	--	0.00001	0.00001	--
<listcomp>	--	1	1	--	0.00000	0.00000	--
<listcomp>	--	1	1	--	0.00000	0.00000	--
_init_next	--	1	1	--	0.00010	0.02809	--
<listcomp>	--	1	1	--	0.00001	0.00001	--
<listcomp>	--	1	1	--	0.00000	0.00000	--
<listcomp>	--	1	1	--	0.00000	0.00000	--
_create_onnx_graphs	--	1	1	--	0.00662	0.02797	--
<listcomp>	--	1	1	--	0.00001	0.00001	--
<listcomp>	--	1	1	--	0.00001	0.00001	--
<listcomp>	--	1	1	--	0.00001	0.00002	--
_provider_name_to_device_type	--	1	1	--	0.00000	0.00000	--
<listcomp>	--	1	1	--	0.00002	0.00002	--
_provider_name_to_device_type	--	7	7	--	0.00000	0.00000	--
<listcomp>	--	1	1	--	0.00000	0.00000	--
_provider_name_to_device_type	--	1	1	--	0.00000	0.00000	--
<listcomp>	--	1	1	--	0.00001	0.00001	--
<listcomp>	--	1	1	--	0.00001	0.00001	--
get_inputs	--	1	1	--	0.00000	0.00000	--
get_outputs	--	1	1	--	0.00000	0.00000	--
__init__	--	2	2	--	0.00004	0.02063	--
get	--	2	2	--	0.00001	0.00004	--
__getitem__	--	2	2	--	0.00001	0.00003	--
encodekey	--	2	2	--	0.00001	0.00002	--
check_str	--	2	2	--	0.00000	0.00000	--
__init__	--	2	2	--	0.00000	0.00000	--
_create_inference_session	--	2	2	--	0.02045	0.02055	--
check_and_nor...rovider_args	--	2	2	--	0.00004	0.00008	--
set_provider_options	--	2	2	--	0.00001	0.00001	--
<dictcomp>	--	2	2	--	0.00000	0.00000	--
<listcomp>	--	2	2	--	0.00000	0.00000	--
<listcomp>	--	2	2	--	0.00000	0.00000	--

load_model	--	2	2	--	0.00001	0.00049	--
_load_bytes	--	2	2	--	0.00002	0.00003	--
inner	--	4	4	--	0.00000	0.00000	--
cast	--	4	4	--	0.00000	0.00000	--
_get_file_path	--	2	2	--	0.00000	0.00000	--
load_model_from_string	--	2	2	--	0.00001	0.00045	--
_deserialize	--	2	2	--	0.00001	0.00044	--
inner	--	2	2	--	0.00000	0.00000	--
cast	--	2	2	--	0.00000	0.00000	--
<method 'Pa...' objects>	--	2	2	--	0.00042	0.00042	--
<built-in method builtins.len>	--	16	16	--	0.00000	0.00000	--
<method 'Serializ...essage' objects>	--	1	1	--	0.00014	0.00014	--
new_instance	--	1	1	--	0.00000	0.00000	--
__init__	--	1	1	--	0.00000	0.00000	--
device_to_providers	--	1	1	--	0.00003	0.00003	--
value	--	100	100	--	0.00003	0.00003	--

```
[49]: import os
if not os.path.exists("mlp_onnx_ort"):
    os.mkdir("mlp_onnx_ort")
train_session2.save_onnx_graph("mlp_onnx_ort")
```

```
[49]: {'model_onnx': 'mlp_onnx_ort\\GradFBOptimizer.model_onnx.onnx',
'learning_rate': {'axpyw_onnx':
'mlp_onnx_ort\\LRateSGDNesterov.learning_rate.axpyw_onnx.onnx'},
'learning_loss': {'loss_grad_onnx':
'mlp_onnx_ort\\SquareLLoss.learning_loss.loss_grad_onnx.onnx',
'loss_score_onnx':
'mlp_onnx_ort\\SquareLLoss.learning_loss.loss_score_onnx.onnx'},
'learning_penalty': {},
'zero_onnx': 'mlp_onnx_ort\\GradFBOptimizer.zero_onnx.onnx',
'train_function': {'_trained_onnx': 'mlp_onnx_ort\\OrtGradientForwardBackwardF
unction_1523278698000.train_function._trained_onnx.onnx',
'_optimized_pre_grad_model': 'mlp_onnx_ort\\OrtGradientForwardBackwardFunction
_1523278698000.train_function._optimized_pre_grad_model.onnx'}}
```

Weights are updated with the following ONNX graph:

```
[50]: %onnxview train_session2.learning_rate.axpyw_onnx_
```

```
[50]: <jyquickhelper.jsipy.render_nb_js_dot.RenderJsDot at 0x162ac3873d0>
```

```
[51]:
```