ONNX Pre-processing WG

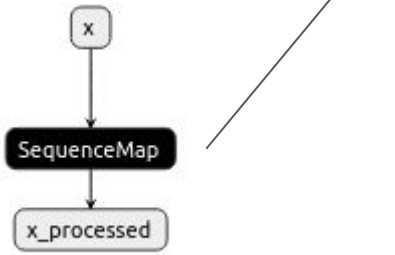
Monthly meeting - Mar 9, 2022

Agenda

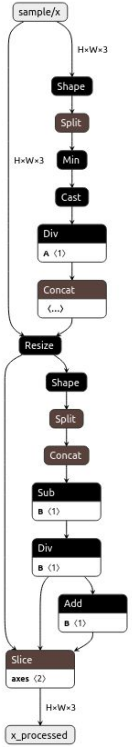
- SequenceMap function - status update
- Using model local functions
- Sequence support in ONNX parser
- Resize and keep aspect ratio semantics
- Open floor

SequenceMap function - status update

https://github.com/onnx/onnx/pull/3892



body



INPUTS

x	name: **x**
	type: **sequence<uint8[H,W,3]>**

OUTPUTS

x_processed	name: **x_processed**
	type: **sequence<uint8[H,W,3]>**

Using model local functions

```
m = parser.parse_model('''
<
  ir_version: 8,
  opset_import: [ "" : 14, "local" : 1],
  producer_name: "test",
  producer_version: "1.0",
  model_version: 1,
  doc_string: "Test preprocessing model",
  metadata_props: [ "preprocessing_fn" : "local.sample_preprocess" ]
>
agraph (uint8[H, W, C] x) => (uint8[H, W, C] y)
{
    x_processed = local.sample_preprocess(x)
    y = Identity(x_processed)
}

<
  opset_import: [ "" : 14 ],
  domain: "local",
  doc_string: "sample preprocessing function"
>
sample_preprocess (x) => (x_processed)
{
    k256f = Constant <value = float[1] {256.0}> ()
    k1f = Constant <value = float[1] {1.0}> ()
    x_shape = Shape (x)
    h, w, c = Split <axis = 0> (x_shape)
    min_extent = Min (h, w)
    min_extent_f = Cast <to = 1> (min_extent)
    ratio_resize = Div (k256f, min_extent_f)
    scales_resize = Concat <axis = 0> (ratio_resize, ratio_resize, k1f)
    roi = Constant <value = float[6] {0.0, 0.0, 0.0, 1.0, 1.0, 1.0}> ()
    x_resized = Resize <mode = \"linear\"> (x, roi, scales_resize)

    k224 = Constant <value = int64[1] {224}> ()
    k2 = Constant <value = int64[1] {2}> ()
    axes = Constant <value = int64[2] {0, 1}> ()
    x_shape2 = Shape (x_resized)
    h2, w2, c2 = Split <axis = 0> (x_shape2)
    hw = Concat <axis = 0> (h2, w2)
    hw_diff = Sub (hw, k224)
    start_xy = Div (hw_diff, k2)
    end_xy = Add (start_xy, k224)
    x_processed = Slice (x_resized, start_xy, end_xy, axes)
}
''')
```

Tagging a single model local function as the preprocessing function.

```
metadata_props: [ "preprocessing_fn" : "local.sample_preprocess" ]
```

Found several issues working with model local functions and onnxruntime and onnx:

Nesting model local functions
https://github.com/microsoft/onnxruntime/issues/10250

Optional inputs within model local function:
https://github.com/microsoft/onnxruntime/issues/10249

Local functions with subgraphs: GraphProto attribute inferencing error
https://github.com/microsoft/onnxruntime/issues/10698

[Solved] Submodel extraction with model local functions
https://github.com/onnx/onnx/issues/3938

Sequence support in ONNX parser

https://github.com/onnx/onnx/issues/4043

How to support sequence inputs/outputs in ONNX parser?

Resize and keep aspect ratio semantics

Example:

Input shape: (853, 1280, 3)

Target size: (200, 200)

resize_not_larger
Output shape: (200, 300, 3)

resize_not_larger
Output shape: (133, 200, 3)

Notes:

- Should we standardize this?

- Would be good to mark dimensions as non-resizable (channels?)

```
resize_not_larger (uint8[H, W, C] x, int64[2] target_size) => (uint8[H, W, C] x_processed)
{
    x_shape = Shape (x)
    h, w, c = Split <axis = 0> (x_shape)
    hw = Concat <axis = 0> (h, w)
    hw_f = Cast <to = 1> (hw)
    target_size_f = Cast <to = 1> (target_size)
    ratios = Div(target_size_f, hw_f)
    ratio_resize = ReduceMin(ratios)

    k1f = Constant <value = float[1] {1.0}> ()
    scales_resize = Concat <axis = 0> (ratio_resize, ratio_resize, k1f)
    x_processed = Resize <mode = \"linear\"> (x, , scales_resize)
}
```

```
resize_not_smaller (uint8[H, W, C] x, int64[2] target_size) => (uint8[H, W, C] x_processed)
{
    x_shape = Shape (x)
    h, w, c = Split <axis = 0> (x_shape)
    hw = Concat <axis = 0> (h, w)
    hw_f = Cast <to = 1> (hw)
    target_size_f = Cast <to = 1> (target_size)
    ratios = Div(target_size_f, hw_f)
    ratio_resize = ReduceMax(ratios)

    k1f = Constant <value = float[1] {1.0}> ()
    scales_resize = Concat <axis = 0> (ratio_resize, ratio_resize, k1f)
    x_processed = Resize <mode = \"linear\"> (x, , scales_resize)
}
```

Open floor