

nn.grid_sample

...

Copy fngrid_sample(X:@Tensor, grid:@Tensor, align_corner:Option, mode:Option, padding_mode:Option,)->Tensor;

...

Given an input X and a flow-field grid, computes the output Y using X values and pixel locations from the grid.

Args

- X
- (@Tensor
-) - Input tensor of shape (N, C, D1, D2, ..., Dr), where N is the batch size, C is the number of channels, D1, D2, ..., Dr are the spatial dimensions.
- grid
- (@Tensor
-) - Input offset of shape (N, D1_out, D2_out, ..., Dr_out, r), where D1_out, D2_out, ..., Dr_out are the spatial dimensions of the grid and output, and r is the number of spatial dimensions. Grid specifies the sampling locations normalized by the input spatial dimensions.
- align_corners
- (Option
-) - default is 0. If align_corners=1, the extrema are considered as referring to the center points of the input's corner pixels. If align_corners=0, they are instead considered as referring to the corner points of the input's corner pixels
- mode
- (Option
-) - default is linear. Three interpolation modes: linear (default), nearest and cubic.
- padding_mode
- (Option
-) - default is zeros. Support padding modes for outside grid values:zeros
- (default),border
- ,reflection
- .
- .

Returns

A Tensor of shape (N, C, D1_out, D2_out, ..., Dr_out) of the sampled values.

Example

...

Copy useorion::operators::nn::NNTrait; useorion::numbers::FixedTrait; useorion::operators::nn::FP16x16NN; useorion::numbers::FP16x16; useorion::operators::tensor::{Tensor,TensorTrait,FP16x16Tensor};

fnexample_grid_sample()->Tensor {

letmutshape=ArrayTrait::new(); shape.append(1); shape.append(2); shape.append(4); shape.append(2);

letmutdata=ArrayTrait::new(); data.append(FP16x16{ mag:655360, sign:true}); data.append(FP16x16{ mag:655360, sign:true}); data.append(FP16x16{ mag:327680, sign:true}); data.append(FP16x16{ mag:327680, sign:true}); data.append(FP16x16{ mag:13107, sign:true}); data.append(FP16x16{ mag:13107, sign:true}); data.append(FP16x16{ mag:655360, sign:false}); data.append(FP16x16{ mag:655360, sign:false}); data.append(FP16x16{ mag:655360, sign:false}); data.append(FP16x16{ mag:13107, sign:true}); data.append(FP16x16{ mag:13107, sign:true}); data.append(FP16x16{ mag:327680, sign:false}); data.append(FP16x16{ mag:327680, sign:false}); data.append(FP16x16{ mag:655360, sign:false}); data.append(FP16x16{ mag:655360, sign:false});

letmutgrid=TensorTrait::new(shape.span(), data.span());

letmutshape=ArrayTrait::new(); shape.append(1); shape.append(1); shape.append(3); shape.append(2);

letmutdata=ArrayTrait::new(); data.append(FP16x16{ mag:0, sign:false}); data.append(FP16x16{ mag:65536, sign:false}); data.append(FP16x16{ mag:131072, sign:false}); data.append(FP16x16{ mag:196608, sign:false}); data.append(FP16x16{ mag:262144, sign:false}); data.append(FP16x16{ mag:327680, sign:false}); letmutX=TensorTrait::new(shape.span(), data.span());

returnNNTrait::grid_sample(@X,@grid, Option::None, Option::None, Option::None,);

```
}  
}  
  
    [[ [0.0000,0.0000,1.7000,0.0000], [0.0000,1.7000,0.0000,0.0000] ] ]  
...
```

[Previous nn.gemm](#) [Next nn.col2im](#)

Last updated 15 days ago