cs133-project

3.3

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Convolutional Class Reference

Class of convolutional layer.

`#include <Convolutional.hpp>`

Inheritance diagram for Convolutional:

Collaboration diagram for Convolutional:

Public Member Functions

- Convolutional ()
  Constructor.
- ~Convolutional ()
  Destructor.
- void init (int cur_in_size, int cur_input_row, int cur_input_col, double node_num, double kernel_row, double kernel_col, double stride_row, double stride_col, std::string padding, std::string name)
  Initialize the layer with args.
- void init (std::vector< std::vector< Eigen::MatrixXd >> kernel, std::vector< double > bias)
  Initialize the layer with given kernel and bias.
- void forward (std::vector< Eigen::MatrixXd > input)
  Spread forward to get the linux.
- int kernel_row () const
  Get the number of rows.
- int kernel_col () const
  Get the number of columns.

Additional Inherited Members

4.1.1 Detailed Description

Class of convolutional layer.

Doing convolution for the input data

Definition at line 12 of file Convolutional.hpp.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Convolutional()

```
Convolutional::Convolutional ( )
```

Constructor.

Definition at line 5 of file Convolutional.hpp.

4.1.2.2 ~Convolutional()

```
Convolutional::~Convolutional ( )
```

Destructor.

Definition at line 8 of file Convolutional.hpp.

4.1.3 Member Function Documentation

4.1.3.1 forward()

```
void Convolutional::forward (
            std::vector< Eigen::MatrixXd > input )  [virtual]
```

Spread forward to get the linux.

Implements Layer.

Definition at line 54 of file Convolutional.hpp.

4.1.3.2 init() [1/2]

```
void Convolutional::init (
            int cur_in_size,
            int cur_input_row,
            int cur_input_col,
            double node_num,
            double kernel_row,
            double kernel_col,
            double stride_row,
            double stride_col,
            std::string padding,
            std::string name )  [virtual]
```

Initialize the layer with args.

Implements Layer.

Definition at line 11 of file Convolutional.hpp.

4.1.3.3 init() [2/2]

```
void Convolutional::init (
            std::vector< std::vector< Eigen::MatrixXd >> kernel,
            std::vector< double > bias )
```

Initialize the layer with given kernel and bias.

Definition at line 47 of file Convolutional.hpp.

4.1.3.4 kernel_col()

```
int Convolutional::kernel_col ( ) const
```

Get the number of columns.

Definition at line 96 of file Convolutional.hpp.

4.1.3.5 kernel_row()

```
int Convolutional::kernel_row ( ) const
```

Get the number of rows.

Definition at line 91 of file Convolutional.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/layers/Convolutional.hpp

4.2 Dense Class Reference

Class of dense layer.

```
#include <dense.hpp>
```

Inheritance diagram for Dense:

Collaboration diagram for Dense:

Public Member Functions

- Dense ()

  Constructor.
- ∼Dense ()

  Destructor.
- void init (int cur_in_size, int cur_input_row, int cur_input_col, double node_num, double kernel_row, double kernel_col, double stride_row, double stride_col, std::string padding, std::string name)

  Initialize the layer with those arguments.
- void init (Eigen::MatrixXd weight, Eigen::MatrixXd bias)

  Initialize the layer with given kernel and bias.
- void forward (std::vector< Eigen::MatrixXd > input)

  Spread forward and get the response.

Additional Inherited Members

4.2.1 Detailed Description

Class of dense layer.

A fully connected layer, just take the input, multiply it by weight and add bias to it and output the result.

Definition at line 15 of file dense.hpp.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Dense()

```
Dense::Dense ( )
```

Constructor.

Definition at line 5 of file dense.hpp.

4.2.2.2 ∼Dense()

```
Dense::∼Dense ( )
```

Destructor.

Definition at line 8 of file dense.hpp.

4.2.3 Member Function Documentation

4.2.3.1 forward()

```
void Dense::forward (
            std::vector< Eigen::MatrixXd > input )  [virtual]
```

Spread forward and get the response.

Implements Layer.

Definition at line 39 of file dense.hpp.

4.2.3.2 init() [1/2]

```
void Dense::init (
            int cur_in_size,
            int cur_input_row,
            int cur_input_col,
            double node_num,
            double kernel_row,
            double kernel_col,
            double stride_row,
            double stride_col,
            std::string padding,
            std::string name )  [virtual]
```

Initialize the layer with those arguments.

Implements Layer.

Definition at line 11 of file dense.hpp.

4.2.3.3 init() [2/2]

```
void Dense::init (
            Eigen::MatrixXd weight,
            Eigen::MatrixXd bias )
```

Initialize the layer with given kernel and bias.

Definition at line 33 of file dense.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/layers/dense.hpp

4.3 Flatten Class Reference

The class of dense layer.

`#include <Flatten.hpp>`

Inheritance diagram for Flatten:

Collaboration diagram for Flatten:

Public Member Functions

- Flatten ()

  Constructor.
- ~Flatten ()

  Destructor.
- void init (int cur_in_size, int cur_input_row, int cur_input_col, double node_num, double kernel_row, double kernel_col, double stride_row, double stride_col, std::string padding, std::string name)

  Initialize the layer with those arguments.
- void forward (std::vector< Eigen::MatrixXd > input)

  Spread forward and get the response.

Additional Inherited Members

4.3.1 Detailed Description

The class of dense layer.

This kind of layer is used to decrease the dimension of the input data, transfer the input data to a single column vector

Definition at line 18 of file Flatten.hpp.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Flatten()

`Flatten::Flatten ( )`

Constructor.

Definition at line 4 of file Flatten.hpp.

4.3.2.2 ~Flatten()

```
Flatten::~Flatten ( )
```

Destructor.

Definition at line 6 of file Flatten.hpp.

4.3.3 Member Function Documentation

4.3.3.1 forward()

```
void Flatten::forward (
            std::vector< Eigen::MatrixXd > input )  [virtual]
```

Spread forward and get the response.

Implements Layer.

Definition at line 30 of file Flatten.hpp.

4.3.3.2 init()

```
void Flatten::init (
            int cur_in_size,
            int cur_input_row,
            int cur_input_col,
            double node_num,
            double kernel_row,
            double kernel_col,
            double stride_row,
            double stride_col,
            std::string padding,
            std::string name )  [virtual]
```

Initialize the layer with those arguments.

Implements Layer.

Definition at line 9 of file Flatten.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/layers/Flatten.hpp

4.4 Identity Class Reference

identity activation function

`#include <identity.hpp>`

Inheritance diagram for Identity:

Collaboration diagram for Identity:

Public Member Functions

- Identity ()
  Default constructor.
- ~Identity ()
  Destructor.
- void init (int cur_in_size, int cur_input_row, int cur_input_col, double node_num, double kernel_row, double kernel_col, double stride_row, double stride_col, std::string padding, std::string name)
  Function of initialize the class.
- void forward (std::vector< Eigen::MatrixXd > input)
  forward spread of the nework

Additional Inherited Members

4.4.1 Detailed Description

identity activation function

As child class of layer

Definition at line 11 of file identity.hpp.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Identity()

`Identity::Identity ( )`

Default constructor.

Constructor.

Definition at line 5 of file identity.hpp.

4.4.2.2 ~Identity()

```
Identity::~Identity ( )
```

Destructor.

Definition at line 8 of file identity.hpp.

4.4.3 Member Function Documentation

4.4.3.1 forward()

```
void Identity::forward (
            std::vector< Eigen::MatrixXd > input )  [virtual]
```

forward spread of the nework

Spread foward to generate the response.

Implements Layer.

Definition at line 27 of file identity.hpp.

4.4.3.2 init()

```
void Identity::init (
            int cur_in_size,
            int cur_input_row,
            int cur_input_col,
            double node_num,
            double kernel_row,
            double kernel_col,
            double stride_row,
            double stride_col,
            std::string padding,
            std::string name )  [virtual]
```

Function of initialize the class.

Initialize the class.

Implements Layer.

Definition at line 12 of file identity.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/activation/identity.hpp

4.5 Layer Class Reference

Layer base class.

`#include <layer.hpp>`

Inheritance diagram for Layer:

Public Types

- enum layerType {
  Conv, Pooling, Dense, Flatten,
  Identity, ReLU, Sigmoid, Softmax }

Public Member Functions

- Layer ()

  default constructor
- virtual ~Layer ()

  virtual deconstructor
- virtual void init (int cur_in_size, int cur_input_row, int cur_input_col, double node_num, double kernel_row, double kernel_col, double stride_row, double stride_col, std::string padding, std::string name)=0

  initialize the parameters (including input size, output size, number of nodes etc.)
- virtual void forward (std::vector< Eigen::MatrixXd > input)=0

  compute the output of this layer
- int node_num () const
- int input_row () const
- int input_col () const
- int output_row () const
- int output_col () const
- std::vector< Eigen::MatrixXd > input () const

  get input
- std::vector< Eigen::MatrixXd > output () const

  get output
- layerType get_type () const

  return the layer type
- std::string get_name () const

  return the layer name
- int in_size () const

  get the size of input
- int out_size () const

  get the size of output

Protected Attributes

- int m_node_num
  number of neurons
- int m_row
- int m_col
- int m_output_row
- int m_output_col
- std::vector< Eigen::MatrixXd > m_input
  input
- std::vector< Eigen::MatrixXd > m_output
  output
- std::string m_name
  layer name
- layerType m_type
  layer type (e.g. conv, pooling, dense)
- int m_in_size
  number of input units of this hidden layers. Equal to the number of output units of the previous layer.
- int m_out_size
  number of output units of this hidden layers. Equal to the number of input units of the next layer.

4.5.1 Detailed Description

Layer base class.

Will be inherited by many child class

Definition at line 15 of file layer.hpp.

4.5.2 Member Enumeration Documentation

4.5.2.1 layerType

`enum Layer::layerType`

=1mm

spread 0pt [l]|X[-1,r]|X[-1,

Enumerator

ConvConvolutional layer.

PoolingPooling Layer.

DenseDence Layer.
FlattenFlatten Layer.
IdentityLayer of Identity activavtion function.
ReLULayer of ReLU activavtion function.
SigmoidLayer of Sigmoid activavtion function.
SoftmaxLayer of Softmax activavtion function.

Definition at line 18 of file layer.hpp.

4.5.3 Constructor & Destructor Documentation

4.5.3.1 Layer()

```
Layer::Layer ( )
```

default constructor

Definition at line 5 of file layer.hpp.

4.5.3.2 ~Layer()

```
Layer::~Layer ( )  [virtual]
```

virtual deconstructor

Definition at line 7 of file layer.hpp.

4.5.4 Member Function Documentation

4.5.4.1 forward()

```
virtual void Layer::forward (
            std::vector< Eigen::MatrixXd > input )  [pure virtual]
```

compute the output of this layer

Implemented in Flatten, Dense, Convolutional, Relu, MaxPooling, Identity, Sigmoid, and Softmax.

4.5.4.2 get_name()

```
std::string Layer::get_name ( ) const
```

return the layer name

Definition at line 54 of file layer.hpp.

4.5.4.3 get_type()

```
Layer::layerType Layer::get_type ( ) const
```

return the layer type

Definition at line 49 of file layer.hpp.

4.5.4.4 in_size()

```
int Layer::in_size ( ) const
```

get the size of input

Definition at line 59 of file layer.hpp.

4.5.4.5 init()

```
virtual void Layer::init (
            int cur_in_size,
            int cur_input_row,
            int cur_input_col,
            double node_num,
            double kernel_row,
            double kernel_col,
            double stride_row,
            double stride_col,
            std::string padding,
            std::string name )  [pure virtual]
```

initialize the parameters (including input size, output size, number of nodes etc.)

Implemented in Flatten, Dense, Relu, Convolutional, MaxPooling, Identity, Sigmoid, and Softmax.

4.5.4.6 input()

```
std::vector< Eigen::MatrixXd > Layer::input ( ) const
```

get input

Definition at line 39 of file layer.hpp.

4.5.4.7 input_col()

```
int Layer::input_col ( ) const
```

Definition at line 22 of file layer.hpp.

4.5.4.8 input_row()

```
int Layer::input_row ( ) const
```

Definition at line 16 of file layer.hpp.

4.5.4.9 node_num()

```
int Layer::node_num ( ) const
```

Definition at line 10 of file layer.hpp.

4.5.4.10 out_size()

```
int Layer::out_size ( ) const
```

get the size of output

Definition at line 64 of file layer.hpp.

4.5.4.11 output()

```
std::vector< Eigen::MatrixXd > Layer::output ( ) const
```

get output

Definition at line 44 of file layer.hpp.

4.5.4.12 output_col()

```
int Layer::output_col ( ) const
```

Definition at line 34 of file layer.hpp.

4.5.4.13 output_row()

```
int Layer::output_row ( ) const
```

Definition at line 28 of file layer.hpp.

4.5.5 Member Data Documentation

4.5.5.1 m_col

```
int Layer::m_col  [protected]
```

Definition at line 89 of file layer.hpp.

4.5.5.2 m_in_size

```
int Layer::m_in_size  [protected]
```

number of input units of this hidden layers. Equal to the number of output units of the previous layer.

Definition at line 107 of file layer.hpp.

4.5.5.3 m_input

```
std::vector<Eigen::MatrixXd> Layer::m_input  [protected]
```

input

Definition at line 95 of file layer.hpp.

4.5.5.4 m_name

```
std::string Layer::m_name  [protected]
```

layer name

Definition at line 101 of file layer.hpp.

4.5.5.5 m_node_num

```
int Layer::m_node_num  [protected]
```

number of neurons

Definition at line 85 of file layer.hpp.

4.5.5.6 m_out_size

```
int Layer::m_out_size  [protected]
```

number of output units of this hidden layers. Equal to the number of input units of the next layer.

Definition at line 110 of file layer.hpp.

4.5.5.7 m_output

```
std::vector<Eigen::MatrixXd> Layer::m_output  [protected]
```

output

Definition at line 98 of file layer.hpp.

4.5.5.8 m_output_col

```
int Layer::m_output_col  [protected]
```

Definition at line 93 of file layer.hpp.

4.5.5.9 m_output_row

```
int Layer::m_output_row  [protected]
```

Definition at line 91 of file layer.hpp.

4.5.5.10 m_row

```
int Layer::m_row  [protected]
```

Definition at line 87 of file layer.hpp.

4.5.5.11 m_type

```
layerType Layer::m_type  [protected]
```

layer type (e.g. conv, pooling, dense)

Definition at line 104 of file layer.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/layer.hpp

4.6 MaxPooling Class Reference

Class of maxpooling class.

`#include <MaxPooling.hpp>`

Inheritance diagram for MaxPooling:

Collaboration diagram for MaxPooling:

Public Member Functions

- MaxPooling ()
  Constructor.
- ~MaxPooling ()
  Destructor.
- void init (int cur_in_size, int cur_input_row, int cur_input_col, double node_num, double kernel_row, double kernel_col, double stride_row, double stride_col, std::string padding, std::string name)
  Initialize the layer with those arguments.
- void forward (std::vector< Eigen::MatrixXd > input)
  Spread forward and get the response.

Additional Inherited Members

4.6.1 Detailed Description

Class of maxpooling class.

This type of layer merge adjucent four data and represent them by the largest one of them

Definition at line 12 of file MaxPooling.hpp.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 MaxPooling()

```
MaxPooling::MaxPooling ( )
```

Constructor.

Definition at line 6 of file MaxPooling.hpp.

4.6.2.2 ~MaxPooling()

```
MaxPooling::~MaxPooling ( )
```

Destructor.

Definition at line 9 of file MaxPooling.hpp.

4.6.3 Member Function Documentation

4.6.3.1 forward()

```
void MaxPooling::forward (
            std::vector< Eigen::MatrixXd > input )  [virtual]
```

Spread forward and get the response.

Implements Layer.

Definition at line 37 of file MaxPooling.hpp.

4.6.3.2 init()

```
void MaxPooling::init (
            int cur_in_size,
            int cur_input_row,
            int cur_input_col,
            double node_num,
            double kernel_row,
            double kernel_col,
            double stride_row,
            double stride_col,
            std::string padding,
            std::string name )  [virtual]
```

Initialize the layer with those arguments.

Implements Layer.

Definition at line 12 of file MaxPooling.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/layers/MaxPooling.hpp

4.7 Net Class Reference

Class of network, consists of several layers.

```
#include <net.hpp>
```

Public Member Functions

- Net ()
  default constructord
- ~Net ()
  default deconstructor
- void init (const std::string &model_path, const std::string &weights_path)
- Eigen::MatrixXd forward (const Eigen::MatrixXd &input)
- void add_layer (Layer *layer)
  add a layer to the network
- void load_model (const std::string &path)
  read model from given path
- void load_weights (const std::string &path)
  read weights from given path
- Eigen::MatrixXd output () const
  return output
- size_t num_layers () const
  return the number of layers

4.7.1 Detailed Description

Class of network, consists of several layers.

Definition at line 23 of file net.hpp.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 Net()

```
Net::Net ( )
```

default constructord

Definition at line 5 of file net.hpp.

4.7.2.2 ~Net()

```
Net::~Net ( )
```

default deconstructor

Definition at line 7 of file net.hpp.

4.7.3 Member Function Documentation

4.7.3.1 add_layer()

```
void Net::add_layer (
            Layer * layer )
```

add a layer to the network

Definition at line 37 of file net.hpp.

4.7.3.2 forward()

```
Eigen::MatrixXd Net::forward (
            const Eigen::MatrixXd & input )
```

compute the ouput of the entire network return the output matrix

Definition at line 21 of file net.hpp.

4.7.3.3 init()

```
void Net::init (
            const std::string & model_path,
            const std::string & weights_path )
```

initialzie all the hidden layers with given the path of model and weights JSON format by default a wrapper of load↩ _model() and load_weights

Definition at line 15 of file net.hpp.

4.7.3.4 load_model()

```
void Net::load_model (
            const std::string & path )
```

read model from given path

Definition at line 42 of file net.hpp.

4.7.3.5 load_weights()

```
void Net::load_weights (
            const std::string & path )
```

read weights from given path

Definition at line 166 of file net.hpp.

4.7.3.6 num_layers()

```
size_t Net::num_layers ( ) const
```

return the number of layers

Definition at line 231 of file net.hpp.

4.7.3.7 output()

```
Eigen::MatrixXd Net::output ( ) const
```

return output

Definition at line 226 of file net.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/net.hpp

4.8 Node Class Reference

Node Class.

```
#include <Node.hpp>
```

Public Member Functions

- Node ()
  donstructor
- ~Node ()
  destructor

4.8.1 Detailed Description

Node Class.

The basic computing unit

Definition at line 11 of file Node.hpp.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 Node()

```
Node::Node ( )
```

donstructor

4.8.2.2 ~Node()

```
Node::~Node ( )
```

destructor

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/Node.hpp

4.9 Relu Class Reference

Activation function of ReLU.

```
#include <ReLU.hpp>
```

Inheritance diagram for Relu:

Collaboration diagram for Relu:

Public Member Functions

- Relu ()
  Constructor.
- ~Relu ()
  Destructor.
- void init (int cur_in_size, int cur_input_row, int cur_input_col, double node_num, double kernel_row, double kernel_col, double stride_row, double stride_col, std::string padding, std::string name)
  Initialize the Layer according to args.
- void forward (std::vector< Eigen::MatrixXd > input)
  Spread forward to generate response.

Additional Inherited Members

4.9.1 Detailed Description

Activation function of ReLU.

ReLU function: f(x) = max(x,0)

Definition at line 12 of file ReLU.hpp.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 Relu()

```
Relu::Relu ( )
```

Constructor.

Definition at line 5 of file ReLU.hpp.

4.9.2.2 ~Relu()

```
Relu::~Relu ( )
```

Destructor.

Definition at line 7 of file ReLU.hpp.

4.9.3 Member Function Documentation

4.9.3.1 forward()

```
void Relu::forward (
            std::vector< Eigen::MatrixXd > input )  [virtual]
```

Spread forward to generate response.

Implements Layer.

Definition at line 24 of file ReLU.hpp.

4.9.3.2 init()

```
void Relu::init (
            int cur_in_size,
            int cur_input_row,
            int cur_input_col,
            double node_num,
            double kernel_row,
            double kernel_col,
            double stride_row,
            double stride_col,
            std::string padding,
            std::string name )  [virtual]
```

Initialize the Layer according to args.

Implements Layer.

Definition at line 10 of file ReLU.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/activation/ReLU.hpp

4.10 Sigmoid Class Reference

Class of Sigmoid activation function.

`#include <Sigmoid.hpp>`

Inheritance diagram for Sigmoid:

Collaboration diagram for Sigmoid:

Public Member Functions

- Sigmoid ()

  Constructor.
- ~Sigmoid ()

  Destructor.
- void init (int cur_in_size, int cur_input_row, int cur_input_col, double node_num, double kernel_row, double kernel_col, double stride_row, double stride_col, std::string padding, std::string name)

  Initialize the Layer according to args.
- void forward (std::vector< Eigen::MatrixXd > input)

  Spread forward to generate response.

Additional Inherited Members

4.10.1 Detailed Description

Class of Sigmoid activation function.

Sigmoid: f(x) = 1/(1 + e^(-x));

Definition at line 11 of file Sigmoid.hpp.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 Sigmoid()

`Sigmoid::Sigmoid ( )`

Constructor.

Definition at line 5 of file Sigmoid.hpp.

4.10.2.2 ~Sigmoid()

```
Sigmoid::~Sigmoid ( )
```

Destructor.

Definition at line 9 of file Sigmoid.hpp.

4.10.3 Member Function Documentation

4.10.3.1 forward()

```
void Sigmoid::forward (
            std::vector< Eigen::MatrixXd > input )  [virtual]
```

Spread forward to generate response.

Implements Layer.

Definition at line 26 of file Sigmoid.hpp.

4.10.3.2 init()

```
void Sigmoid::init (
            int cur_in_size,
            int cur_input_row,
            int cur_input_col,
            double node_num,
            double kernel_row,
            double kernel_col,
            double stride_row,
            double stride_col,
            std::string padding,
            std::string name )  [virtual]
```

Initialize the Layer according to args.

Implements Layer.

Definition at line 12 of file Sigmoid.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/activation/Sigmoid.hpp

4.11 Softmax Class Reference

Class of Softmax activation function.

`#include <softmax.hpp>`

Inheritance diagram for Softmax:

Collaboration diagram for Softmax:

Public Member Functions

- Softmax ()

  Constructor.
- ~Softmax ()

  Destructor.
- void init (int cur_in_size, int cur_input_row, int cur_input_col, double node_num, double kernel_row, double kernel_col, double stride_row, double stride_col, std::string padding, std::string name)

  Initialize the Layer according to args.
- void forward (std::vector< Eigen::MatrixXd > input)

  Spread forward to generate response.

Additional Inherited Members

4.11.1 Detailed Description

Class of Softmax activation function.

f(xi) = (e^(xi))/(sigma e^(xj))

Definition at line 11 of file softmax.hpp.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 Softmax()

`Softmax::Softmax ( )`

Constructor.

Definition at line 4 of file softmax.hpp.

4.11.2.2 ~Softmax()

```
Softmax::~Softmax ( )
```

Destructor.

Definition at line 6 of file softmax.hpp.

4.11.3 Member Function Documentation

4.11.3.1 forward()

```
void Softmax::forward (
            std::vector< Eigen::MatrixXd > input )  [virtual]
```

Spread forward to generate response.

Implements Layer.

Definition at line 23 of file softmax.hpp.

4.11.3.2 init()

```
void Softmax::init (
            int cur_in_size,
            int cur_input_row,
            int cur_input_col,
            double node_num,
            double kernel_row,
            double kernel_col,
            double stride_row,
            double stride_col,
            std::string padding,
            std::string name )  [virtual]
```

Initialize the Layer according to args.

Implements Layer.

Definition at line 9 of file softmax.hpp.

The documentation for this class was generated from the following file:

- /home/martin/CS133/Project/cs133-project/include/activation/softmax.hpp

Chapter 5

File Documentation

5.1 /home/martin/CS133/Project/cs133-project/include/activation/identity.hpp File Reference

```
#include "../Layer.hpp"
#include "identity.hpp"
```

Include dependency graph for identity.hpp:

5.2 /home/martin/CS133/Project/cs133-project/include/activation/identity.hpp File Reference

```
#include "../Layer.hpp"
#include "identity.hpp"
```

Include dependency graph for identity.hpp:

5.3 /home/martin/CS133/Project/cs133-project/include/activation/ReLU.hpp File Reference

```
#include "../Layer.hpp"
#include "reLU.hpp"
```

Include dependency graph for ReLU.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Relu

  Activation function of ReLU.

5.4 /home/martin/CS133/Project/cs133-project/include/activation/ReLU.hpp File Reference

```
#include "../Layer.hpp"
#include "reLU.hpp"
```

Include dependency graph for ReLU.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Relu

  Activation function of ReLU.

5.5 /home/martin/CS133/Project/cs133-project/include/activation/Sigmoid.hpp File Reference

```
#include "../Layer.hpp"
#include "sigmoid.hpp"
```

Include dependency graph for Sigmoid.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Sigmoid

  Class of Sigmoid activation function.

5.6 /home/martin/CS133/Project/cs133-project/include/activation/Sigmoid.hpp File Reference

```
#include "../Layer.hpp"
#include "sigmoid.hpp"
```

Include dependency graph for Sigmoid.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Sigmoid

  Class of Sigmoid activation function.

5.7 /home/martin/CS133/Project/cs133-project/include/activation/softmax.hpp File Reference

```
#include "../Layer.hpp"
#include "softmax.hpp"
```

Include dependency graph for softmax.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Softmax

  Class of Softmax activation function.

5.8 /home/martin/CS133/Project/cs133-project/include/activation/softmax.hpp File Reference

```
#include "../Layer.hpp"
#include "softmax.hpp"
```

Include dependency graph for softmax.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Softmax

  Class of Softmax activation function.

5.9 /home/martin/CS133/Project/cs133-project/include/layer.hpp File Reference

```
#include <Eigen/Core>
#include <vector>
#include <cstddef>
#include <string>
#include "layer.hpp"
```

Include dependency graph for layer.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Layer

  Layer base class.

5.10 /home/martin/CS133/Project/cs133-project/include/layer.hpp File Reference

```
#include <Eigen/Core>
#include <vector>
#include <cstddef>
#include <string>
#include "layer.hpp"
```

Include dependency graph for layer.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Layer

  Layer base class.

5.11 /home/martin/CS133/Project/cs133-project/include/layers/Convolutional.hpp File Reference

```
#include "../Layer.hpp"
#include "convolutional.hpp"
```

Include dependency graph for Convolutional.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Convolutional

  Class of convolutional layer.

5.12 /home/martin/CS133/Project/cs133-project/include/layers/Convolutional.hpp File Reference

```
#include "../Layer.hpp"
#include "convolutional.hpp"
```

Include dependency graph for Convolutional.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Convolutional

  Class of convolutional layer.

5.13 /home/martin/CS133/Project/cs133-project/include/layers/dense.hpp File Reference

```
#include "../Layer.hpp"
#include "dense.hpp"
```

Include dependency graph for dense.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Dense

  Class of dense layer.

5.14 /home/martin/CS133/Project/cs133-project/include/layers/dense.hpp File Reference

```
#include "../Layer.hpp"
#include "dense.hpp"
```

Include dependency graph for dense.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Dense

  Class of dense layer.

5.15 /home/martin/CS133/Project/cs133-project/include/layers/Flatten.hpp File Reference

```
#include "../Layer.hpp"
#include "flatten.hpp"
```

Include dependency graph for Flatten.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Flatten

  The class of dense layer.

5.16 /home/martin/CS133/Project/cs133-project/include/layers/Flatten.hpp File Reference

```
#include "../Layer.hpp"
#include "flatten.hpp"
```

Include dependency graph for Flatten.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Flatten

  The class of dense layer.

5.17 /home/martin/CS133/Project/cs133-project/include/layers/MaxPooling.hpp File Reference

```
#include "../Layer.hpp"
#include "maxPooling.hpp"
```

Include dependency graph for MaxPooling.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class MaxPooling

  Class of maxpooling class.

5.18 /home/martin/CS133/Project/cs133-project/include/layers/MaxPooling.hpp File Reference

```
#include "../Layer.hpp"
#include "maxPooling.hpp"
```

Include dependency graph for MaxPooling.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class MaxPooling

  Class of maxpooling class.

5.19 /home/martin/CS133/Project/cs133-project/include/net.hpp File Reference

```
#include "Layer.hpp"
#include "layers/Convolutional.hpp"
#include "layers/Dense.hpp"
#include "layers/MaxPooling.hpp"
#include "layers/Flatten.hpp"
#include "activation/ReLU.hpp"
#include "activation/Sigmoid.hpp"
#include "activation/Softmax.hpp"
#include "../third_party/toy_json/src/toy_json.cpp"
#include <Eigen/Core>
#include <cassert>
#include "net.hpp"
```

Include dependency graph for net.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Net

  Class of network, consists of several layers.

5.20 /home/martin/CS133/Project/cs133-project/include/net.hpp File Reference

```
#include "Layer.hpp"
#include "layers/Convolutional.hpp"
#include "layers/Dense.hpp"
#include "layers/MaxPooling.hpp"
#include "layers/Flatten.hpp"
#include "activation/ReLU.hpp"
#include "activation/Sigmoid.hpp"
#include "activation/Softmax.hpp"
#include "../third_party/toy_json/src/toy_json.cpp"
#include <Eigen/Core>
#include <cassert>
#include "net.hpp"
```

Include dependency graph for net.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class Net

  Class of network, consists of several layers.

5.21 /home/martin/CS133/Project/cs133-project/include/Node.hpp File Reference

Class of Node.

```
#include <Eigen/Core>
```

Include dependency graph for Node.hpp:

Classes

- class Node

  Node Class.

5.21.1 Detailed Description

Class of Node.