Documentation of a fully-connected neural network

This is a documentation about a fully connected neural network built in Ptolemy II. The purpose of this document is to describe the actors that have been used, specify the input data types and define its functionality.

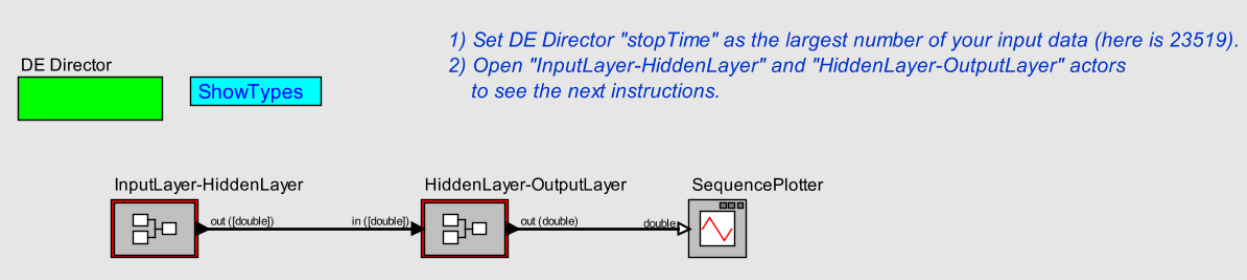

Image-1: Complete fully connected neural network in Ptolemy II.

Image-1 shows the fully connected neural network built in Ptolemy II platform. There are three actors here: *InputLayer-HiddenLayer*, *HiddenLayer-OutputLayer* (composite actors) and a *SequencePlotter* actor to display the results.

InputLayer-HiddenLayer

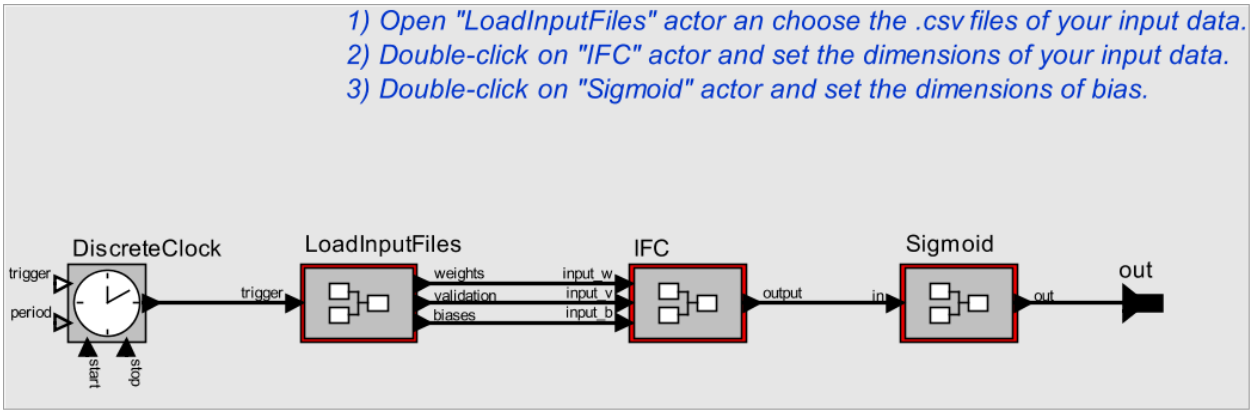

Image-2: InputLayer-HiddenLayer composite actor.

This composite actor reads the input data, do the mathematics between matrices and gives the output between the input layer and the hidden layer.

1. ***DiscreteClock***
   This actor is used to trigger *LineReader* actors in order to read the input data.

2. ***LoadInputFiles***

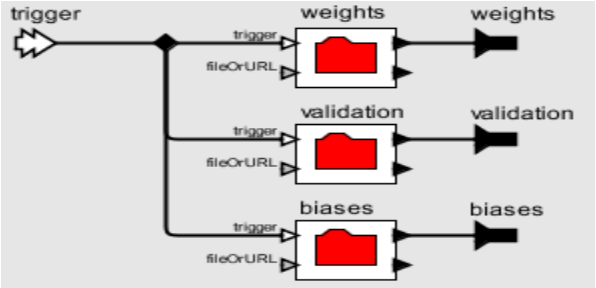

Image-3: LoadInputFiles composite actor.

This is a composite actor that is used to read the input data files. *Double-click* on weights, validation, biases *LineReader* actors to choose the input data files. Input data files (weights, validation, biases) must be *column-vectors*. This means that, a *(n x m)* matrix must be converted to a *((n x m) x 1)* vector, by turning every row into column.
The actor has one input port (*trigger)* and 3 output ports (*weights, validation, biases).*

3. ***IFC (InputFullyConnected)***

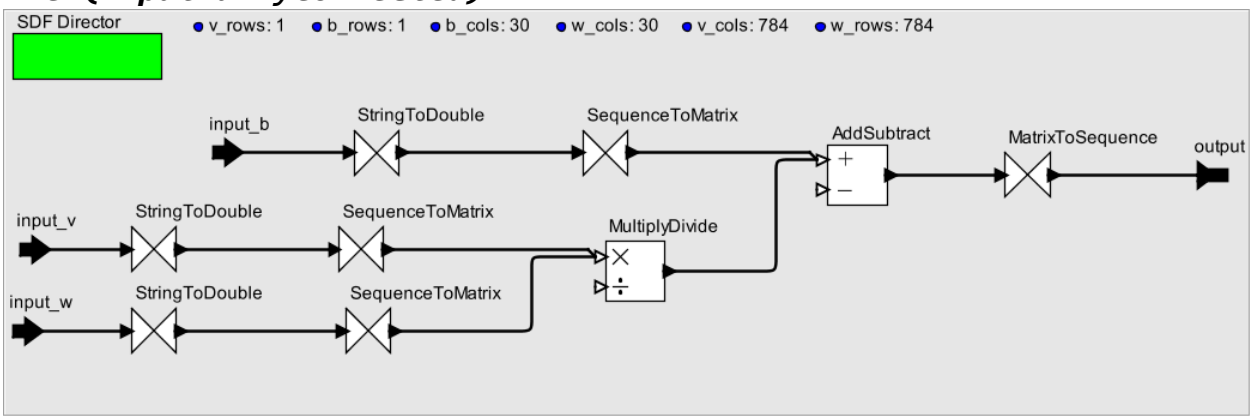

Image-4: IFC composite actor.

The mathematical part of the network is implemented here. *Double-click* on the *IFC* actor to write the dimensions of the input data, which should satisfy the multiplication rules.

This actor has 3 input ports (input data) and one output port (the validation data for the next layer).

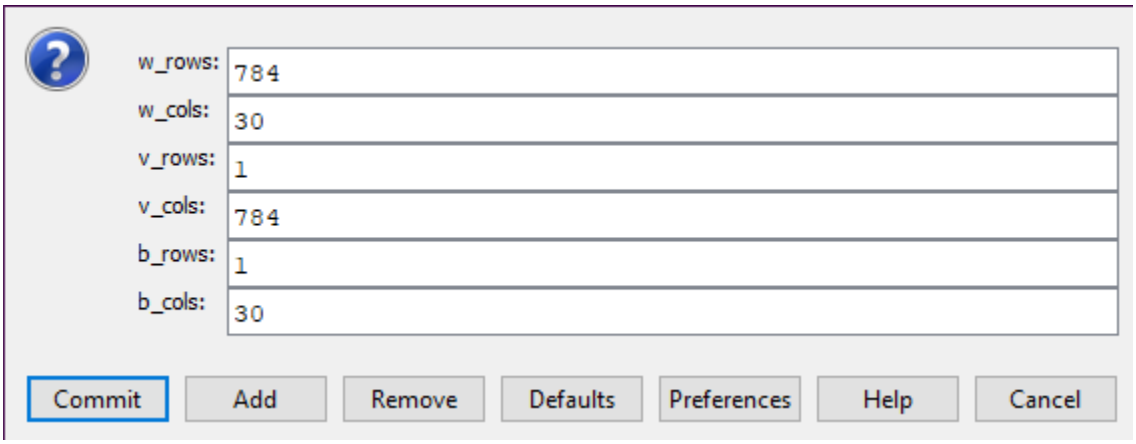

Image-5: IFC composite actor (double-click).

4. Sigmoid

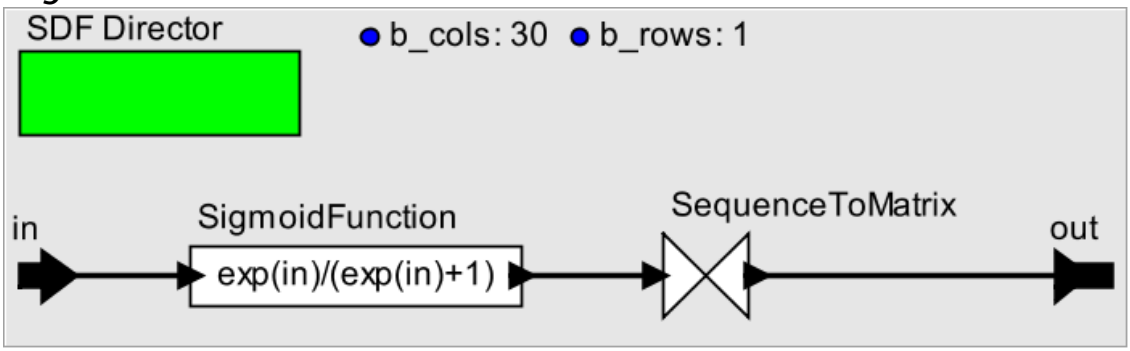

Image-6: Sigmoid composite actor.

This actor is used to apply the *activation function* on the output data. Here, we have *sigmoid* activation function. *Double-Click* on the on the *Sigmoid* actor to write the dimensions of the output data (same dimensions as biases, that's why *b_cols*, *b_rows* are used).

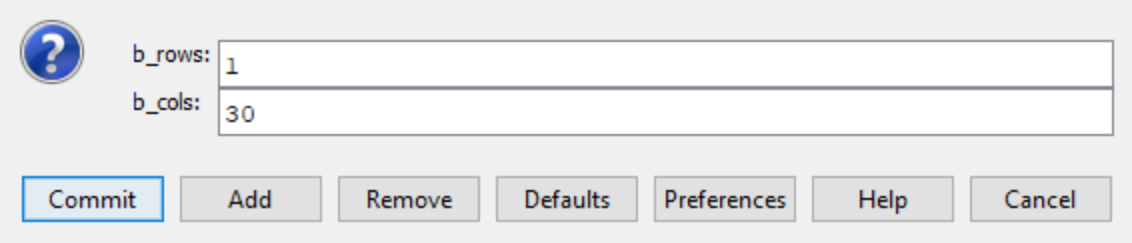

Image-7: Sigmoid composite actor (double-click).

HiddenLayer-OutputLayer

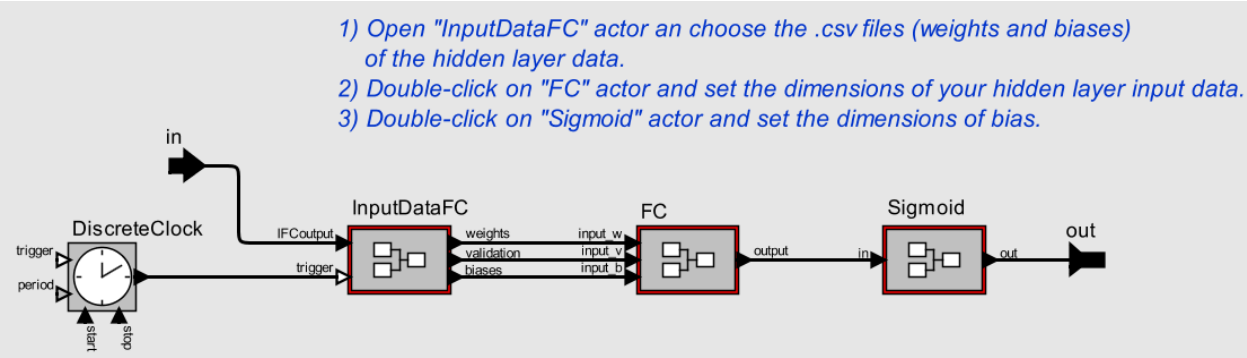

Image-8: HiddenLayer-OutputLayer composite actor.

Same as the *InputLayer-OutputLayer* composite actor, but this time for the connection between hidden layer and the output layer.

1. ***DiscreteClock***
   Same as before.

2. ***InputDataFC***

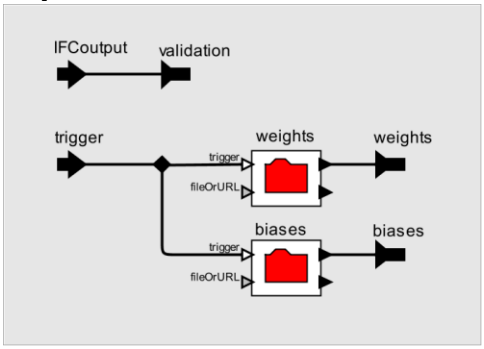

Image-9: InputDataFC composite actor.

Here, the weights and biases are read, but for the *HiddenLayer-OutputLayer* actor. There is no validation LineReader actor this time, because the output of the *InputLayer-OutputLayer* comes in as input here.

3. FC (Fully Connected)

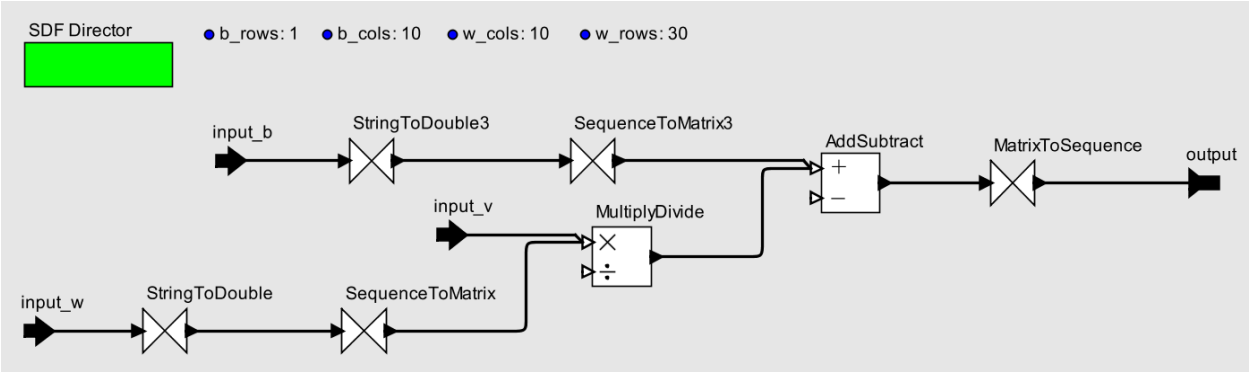

Image-10: FC composite actor.

Same as before, *double-click* and write the dimensions of the input data (this time only for weights and biases).

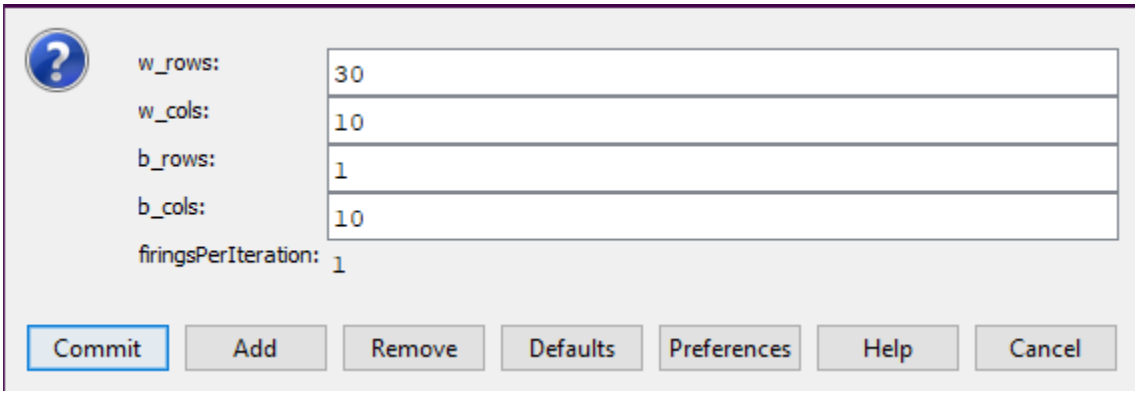

Image-11: FC composite actor (double click).

4. Sigmoid

Same as before, but with the dimensions of the new biases (connection between hidden layer and output layer).

NOTE: *SequenceToMatrix* actor inside *Sigmoid* actor is deleted, in order to display the results as sequences in a *SequencePlotter* actor.

NOTES

1. This Ptolemy II network works only in DE Domain. With some modifications it can also work in SDF Domain.
2. You can use more *HiddenLayer-OutputLayer* composite actors, if you want more hidden layers in your network.
3. You can also use another activation function. Just modify the mathimatical equation in *Expression* actor, inside the *Sigmoid* composite actor.
4. The .csv files format must be CSV (Comma Delimited).
5. Except from .csv, .txt format type of input data is compatible too.
6. In this repository, there is also the userLibrary, which contains these models. Copy-paste the userLibrary.xml file in the userLibrary destination folder and open it with Ptolemy II.