The nndraw package

Carlos Cardoso Dias
carlosdias.dev@gmail.com

1.0 from 2021/06/22

1 Introduction

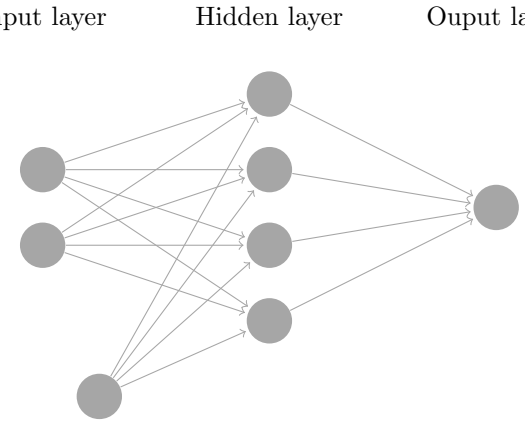
nndraw is a LaTeXpackage which provides utilities to draw neural networks in an efficient way. Currently this package provides utilities to draw fully connected feedforward neural networks with an arbitrary number of layers described inside the 'fullyconnectednn' environment using the command 'nnlayer'.

2 Usage

An example of usage is shown below, in this example, a fully connected feedforward deep neural network is provided with two inputs in the first layer, one hidden layer with four neurons and one output layer with one output. This example shows how easy it is to customize the presence/absence of biases in any layer as well as its position.

```
\begin{fullyconnectednn}[biasy = -4.5,
                         titley = 1,
                         inout = false]
    \nnlayer[title = Input layer]{2}
    \nnlayer[title = Hidden layer]{4}
    \nnlayer[title = Ouput layer, hasbias = false]{1}
\end{fullyconnectednn}
```

Which is drawn as



3 Implementation

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{nndraw}[2021/06/20 Neural Networks Draw LaTeX package]
\RequirePackage{tikz}
\pgfdeclarelayer{back}
\pgfsetlayers{back,main}
```

`\nnlayerNoText`

```
\newcommand\nnlayerNoText[1]{}

\newif\ifnnlayerHasBias
\pgfkeys{
 /nnlayer/.is family, /nnlayer,
 default/.style = {title = , text = \nnlayerNoText, hasbias = true, bias = , color = , biascolo
 title/.estore in = \nnlayerTitle,
 text/.store in = \nnlayerText,
 hasbias/.is if = nnlayerHasBias,
 bias/.estore in = \nnlayerBias,
 color/.estore in = \nnlayerColor,
 biascolor/.estore in = \nnlayerBiasColor,
}
\newcounter{numlayers}
\newcounter{nninputlayer}
\newcounter{lastnnsize}
\tikzstyle{neuron}=[circle, minimum size=6mm, fill=gray!70]
\tikzstyle{neuron_connection}=[->, shorten >=1pt, gray!70]
```

`\nnlayer`

```
\newcommand\nnlayer[2][]{
    \pgfkeys{/nnlayer, default, #1}
    \node[text width = \fullyconnectednnTextWidth, align = center] at (\fullyconnectednnLayerse
    \thenumlayers,\fullyconnectednnTitleY) {\nnlayerTitle};
    \if\thenninputlayer0
```

```
        \setcounter{nninputlayer}{#2}
    \fi
    \foreach \i in {1,...,#2}
    {
        \node[neuron, yshift=(#2 - \thenninputlayer)*5 mm, fill=\nnlayerColor] (\thenumlayers-\
    }
    \if\thelastnnsize0
    \else
        \ifnnlayerHasBias
            \node[neuron, yshift=5mm, fill=\nnlayerBiasColor] (bias-\thenumlayers) at (\thenuml
        \fi
        \begin{pgfonlayer}{back}
```

node connections

```
            \foreach \i in {1,...,\thelastnnsize}
            {
                \foreach \j in {1,...,#2}
                {
                    \draw[neuron_connection] (\the\numexpr\thenumlayers-1\relax-\i) -- (\thenum
                }
            }
```

bias connections

```
            \ifnnlayerHasBias
                \foreach \j in {1,...,#2}
                {
                    \draw[neuron_connection] (bias-\thenumlayers) -- (\thenumlayers-\j);
                }
            \fi
        \end{pgfonlayer}
    \fi
    \setcounter{lastnnsize}{#2}
    \stepcounter{numlayers}
}

\newif\iffullyconnectednnInout
\pgfkeys{
 /fullyconnectednn/.is family, /fullyconnectednn,
 default/.style = {input = \nnlayerNoText, output = \nnlayerNoText, layersep = 3, biasx = 0.75,
 input/.store in = \fullyconnectednnInput,
 output/.store in = \fullyconnectednnOutput,
 layersep/.estore in = \fullyconnectednnLayersep,
 biasx/.estore in = \fullyconnectednnBiasX,
 biasy/.estore in = \fullyconnectednnBiasY,
 titley/.estore in = \fullyconnectednnTitleY,
 inout/.is if = fullyconnectednnInout,
 text width/.estore in = \fullyconnectednnTextWidth,
}
```

`fullyconnectednn`

```
\newenvironment{fullyconnectednn}[1][]
```

```
    {
    \pgfkeys{/fullyconnectednn, default, #1}
    \begin{center}
    \begin{tikzpicture}
    \setcounter{numlayers}{0}
    \setcounter{nninputlayer}{0}
    \setcounter{lastnnsize}{0}
    }
    {
    \iffullyconnectednnInout
    \begin{pgfonlayer}{back}
```

Inputs

```
    \foreach \i in {1,...,\thenninputlayer}
    {
        \draw[<-, shorten <=1pt] (0-\i) -- ++(-1,0) node[left]{\fullyconnectednnInput{\i}};
    }

```

Outputs

```
    \foreach \i in {1,...,\thelastnnsize}
    {
        \draw[->, shorten <=1pt] (\the\numexpr\thenumlayers-1\relax-\i) -- ++(1,0) node[right]{
    }
    \end{pgfonlayer}
    \fi
    \end{tikzpicture}
    \end{center}
    }
```