TikZ Tutorial

John Tanner

27 March 2025

1 Introduction

TikZ is a powerful and flexible package for creating high-quality graphics which can be used either within LaTeX documents, or to produce nice .pdf files using `\documentclass{standalone}`. While other graphics packages may offer drag-and-drop interfaces, TikZ is entirely code-based, meaning you (very manually) define the objects and relationships through commands, giving you an exceptional level of customisation. Its vast range of tools and libraries makes it suitable for anything from simple geometric illustrations to more sophisticated flowcharts and diagrams. It can also be used for data visualisation, but we won't cover data visualisation here, and I would personally recommend doing data visualisation tasks using something like Matplotlib in Python anyway. Now, TikZ may initially feel like it has a steep learning curve, but once you get used to some of the syntax it becomes quite natural, and in my opinion the level of control that TikZ provides justifies the initial learning curve.

This file contains a collection of examples and exercises showing how to use the basic features of TikZ to draw and customise shapes, diagrams, and text in various ways. Specifically, we will demonstrate and use two basic, but very useful, TikZ commands:

- `\draw[<options>] <commands>;`
  - used for creating and customising simple geometric shapes like lines, circles, and polygons, or more complex paths using a combination of coordinates and drawing options.
- `\node[<options>] (<label>) at (<coordinates>) {<text>};`
  - used to place text, such as labels or annotations, inside various shapes like circles or rectangles, at given coordinates or relative positions in a diagram.
- *Note:* Supplying `[<options>]` for `\draw` is optional, and supplying `[<options>]` and `(<label>)` for `\node` are both optional too.

More useful infomation about the various libraries and option in TikZ can be found at https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ (as well as in a

plethora of other useful online resources, like Youtube videos, TeX StackExchange, and of course ChatGPT).

Contents

1 Introduction **1**

2 Examples **3**
2.1 The coordinate systems in TikZ 3
2.2 Drawing some shapes 4
2.3 Styling the shapes 5
2.4 Drawing a basic tree diagram 6
2.5 Styling the tree diagram 7
2.6 Drawing a basic neural network 8
2.7 Styling the neural network 9
2.8 Using `\includegraphics` in a `\node` 10
2.9 Drawing a flowchart 11
2.10 Drawing a particle in a magnetic field being controlled by a laser 12

3 Closing remarks **13**

4 Exercises **14**
4.1 Drawing a happy face 14
4.2 Drawing a colourful envelope 15
4.3 Turn a 1-star rating into a 5-star rating. 16
4.4 Drawing a flowchart 17
4.5 Drawing a convolutional neural network 18

2 Examples

2.1 The coordinate systems in TikZ

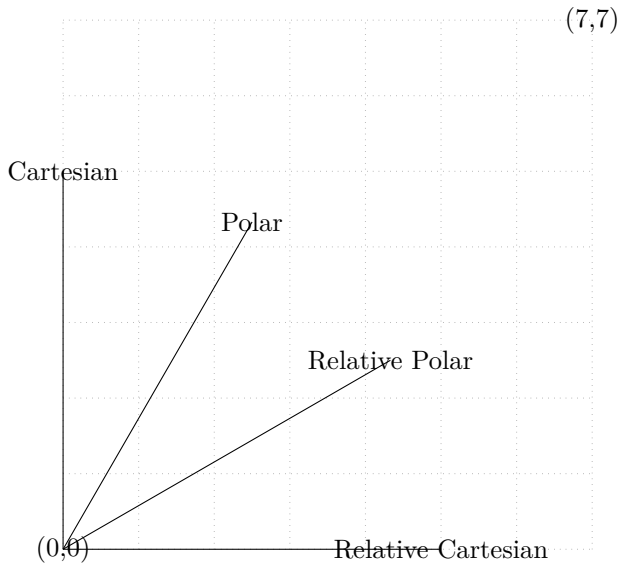


Figure 1: Different ways of specifying coordinates in TikZ.

2.2 Drawing some shapes

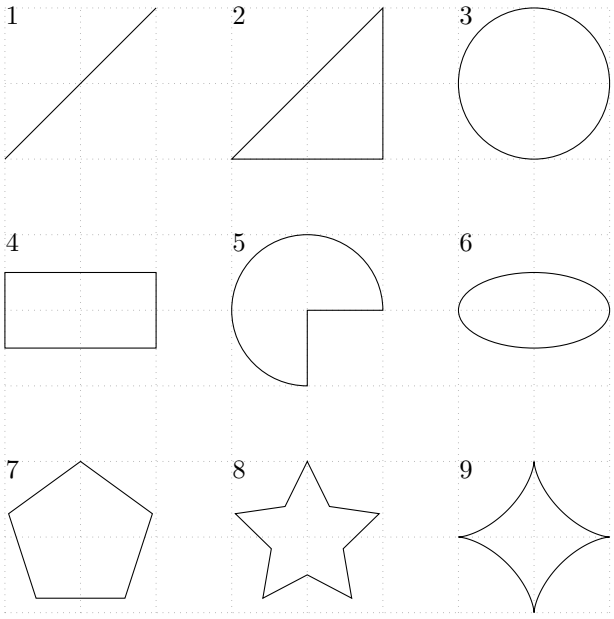


Figure 2: Standard shapes.

2.3 Styling the shapes

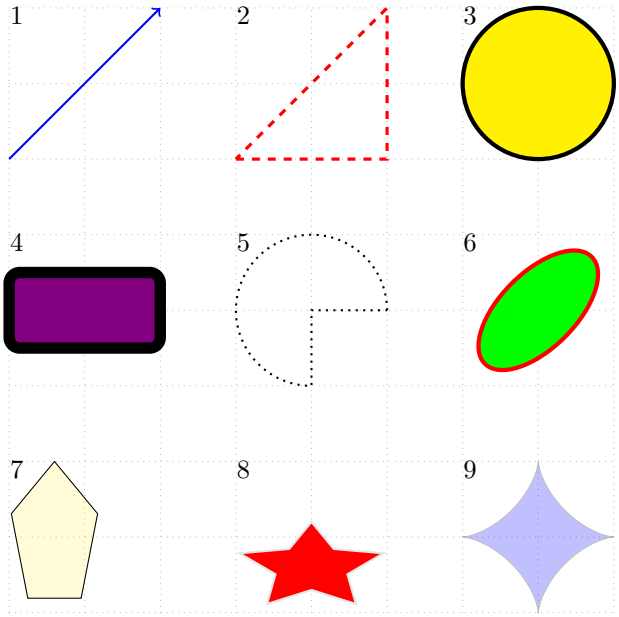


Figure 3: Standard shapes with custom style options.

2.4 Drawing a basic tree diagram

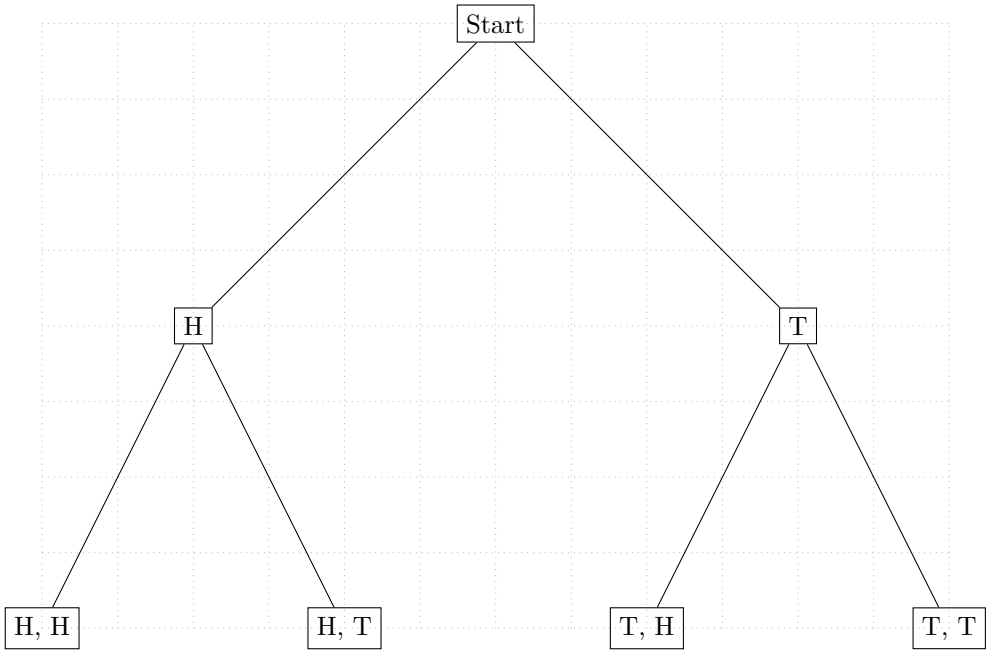


Figure 4: A basic tree diagram illustrating the possible outcomes of two coin flips.

2.5 Styling the tree diagram

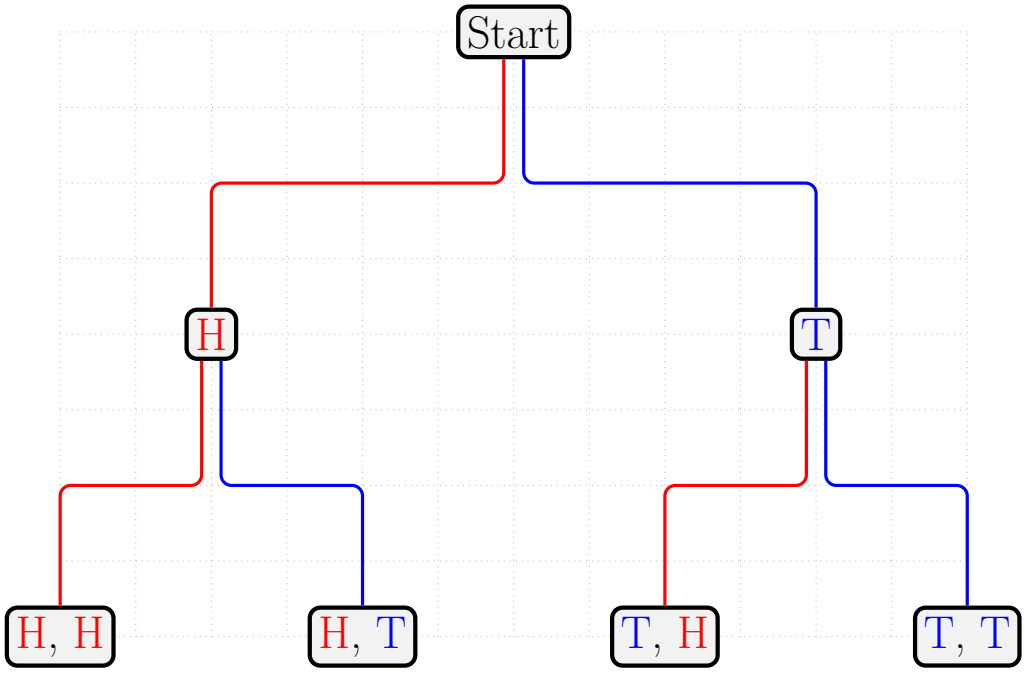


Figure 5: A styled tree diagram illustrating the possible outcomes of two coin flips.

2.6 Drawing a basic neural network

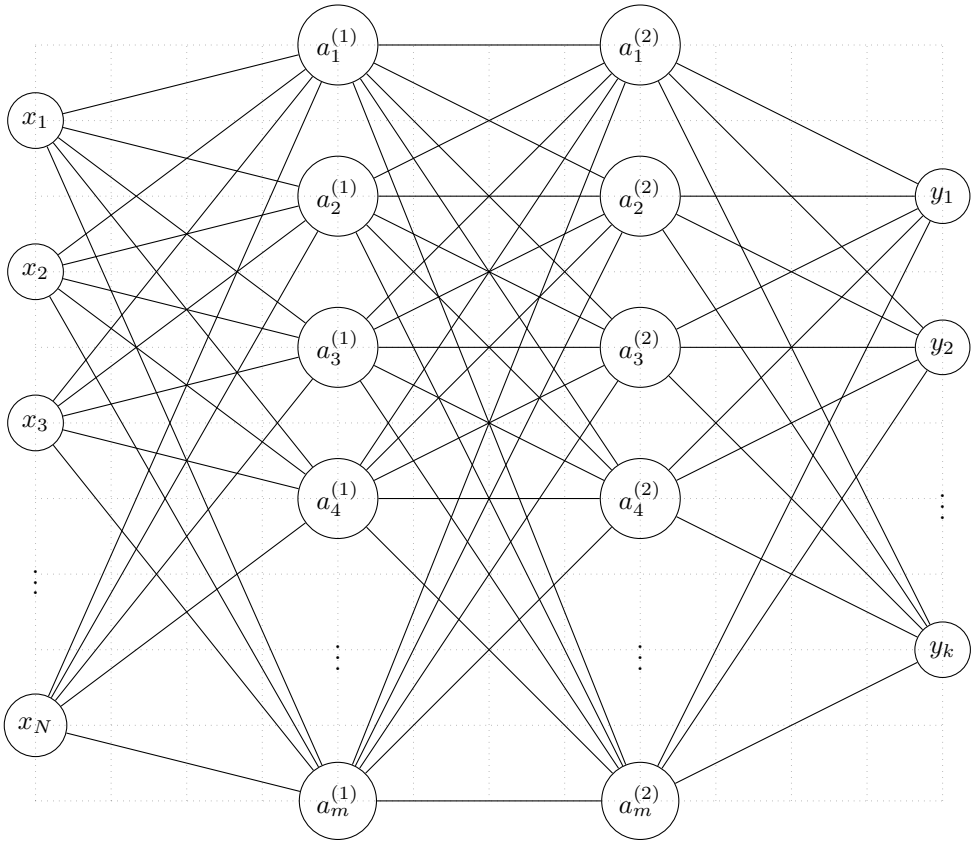


Figure 6: A neural network.

2.7 Styling the neural network

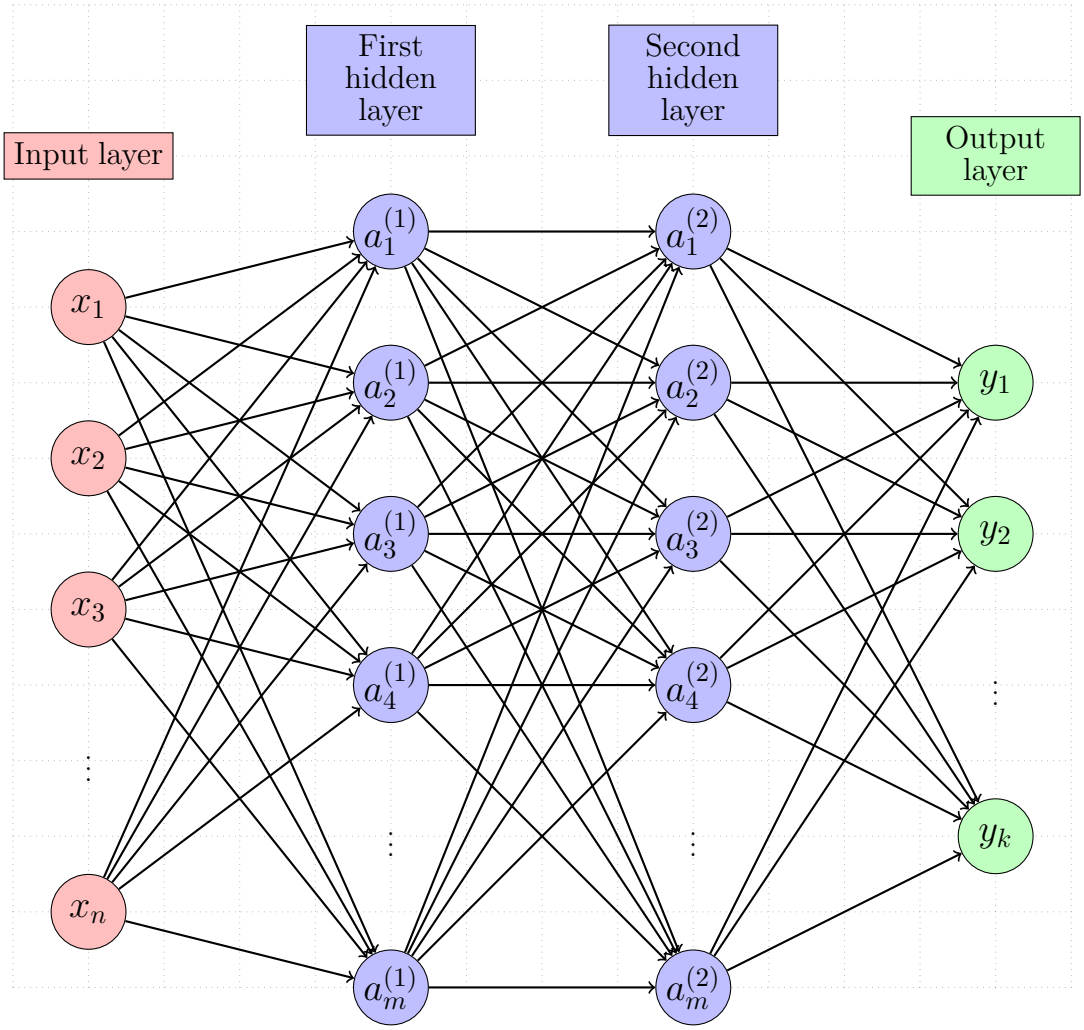


Figure 7: A neural network with colour, larger text, evenly sized neurons, thick arrows instead of plain lines, and labelled layers containing wrapped text.

2.8 Using \includegraphics in a \node

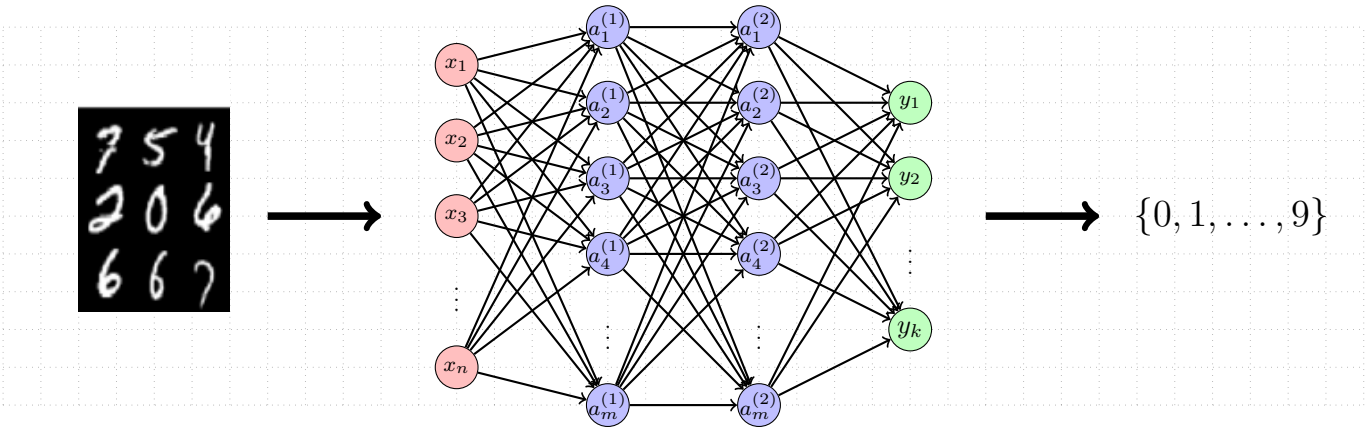


Figure 8: A basic diagram illustrating the task of multi-class classification using the handwritten digits in the MNIST dataset and a neural network.

2.9 Drawing a flowchart

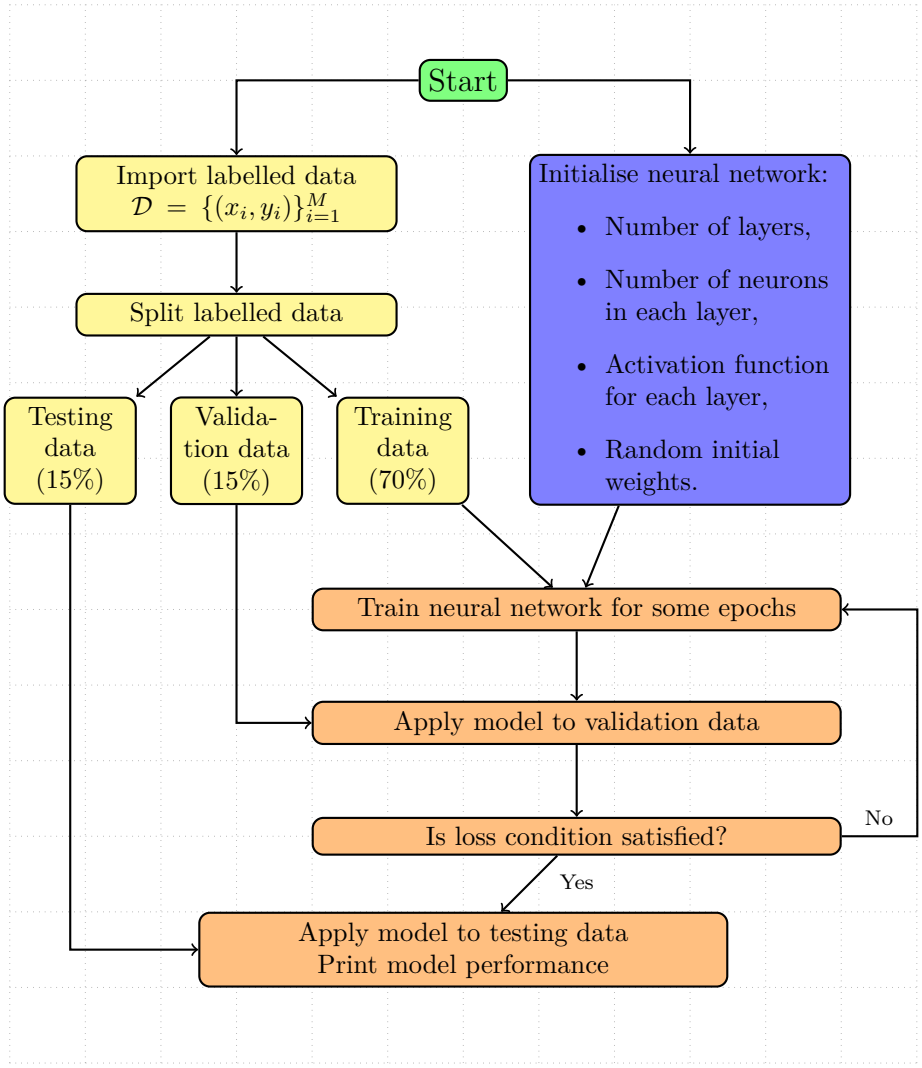


Figure 9: A flowchart describing the workflow involved with applying a neural network to some labelled data for the purposes of supervised machine learning.

2.10 Drawing a particle in a magnetic field being controlled by a laser

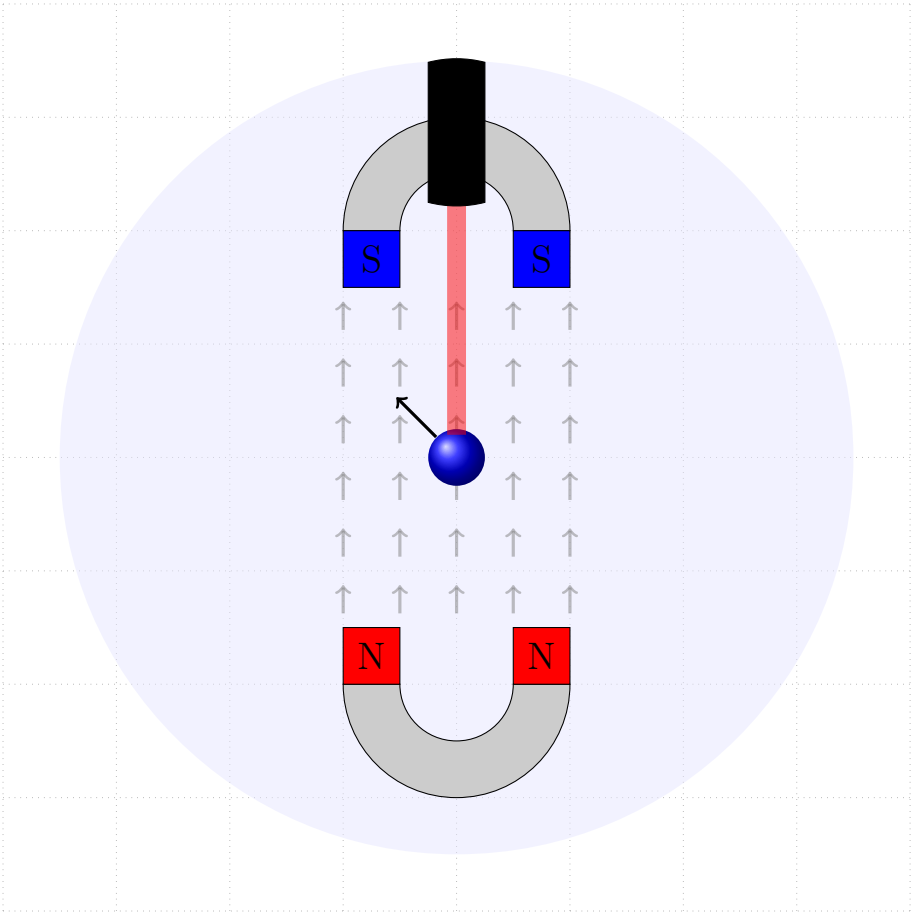


Figure 10: A spin-$\frac{1}{2}$ particle in a magnetic field being manipulated with a tunable laser.

3 Closing remarks

In this tutorial we have seen a few different ways of producing shapes with some text inside, so I thought I would just take a moment to give you my perspective on the different methods.

1. **Using one `\node`:**
   This method refers to using
   `\node[draw] (label) at (x,y) {Text}`,
   possibly with some other options, to produce both the shape and the text. This is the most convenient option, since we can automatically connect lines between the node and other things without needing to write out the explicit coordinates. However I find that sometimes it can annoying to try and force the shape to format in consistent ways, often changing size pretty significantly in response to small changes in text (see Example 6).

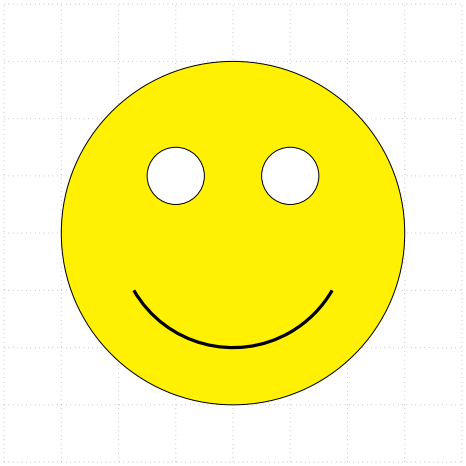
2. **Using two `\node`'s:**
   This method refers to using
   `\node[draw, inner sep=5mm] (label) at (x,y) {};`
   `\node at (x,y) {Text}`,
   possibly with other options, where the first `\node` produces a shape of fixed size (note that there exists options other than `inner sep` which can fix the size of a `\node`), and the second `\node` produces text at the same location as the first. This is convenient if you are experiencing something like when the size of a node is unnecessarily changing in response to small changes in the text and you just want to force the nodes to have the same size while maintaining the ability to automatically reference the node using its label. However I personally find that something its annoying to customise the format of the shape drawn by the `\node`.

3. **Using one `\draw` and one `\node`:**
   This method refers to using
   `\draw[<options>] <commands>;`
   `\node at (x,y) {Text}`,
   possibly with other options, where the `\draw` draws a shape, and the `\node` produces some text within the shape. This is the most customisable option, but also the most manual, and using it means that we can no longer automatically draw lines between the node and other objects without explicitly specifying some coordinates. We can choose to label the `\node` using `\node (label) at (x,y) {Text}`, but referencing the label to draw a line to another object won't necessarily line up the end point of the line with the edge of the shape the we drew around the text. We can work around this though by using relative coordinates to tinker with the end point of the line such as in `\draw (label) ++(x,y) -- …`.

4 Exercises

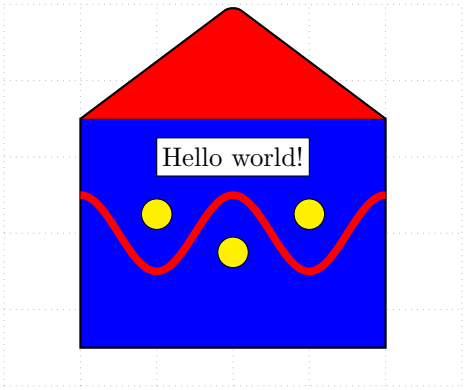
4.1 Drawing a happy face

Draw the happy face below using TikZ.



4.2 Drawing a colourful envelope

Draw the colourful envelope below using TikZ.
Hint: Look at the code for Example 2.1 used to produce shape 9.

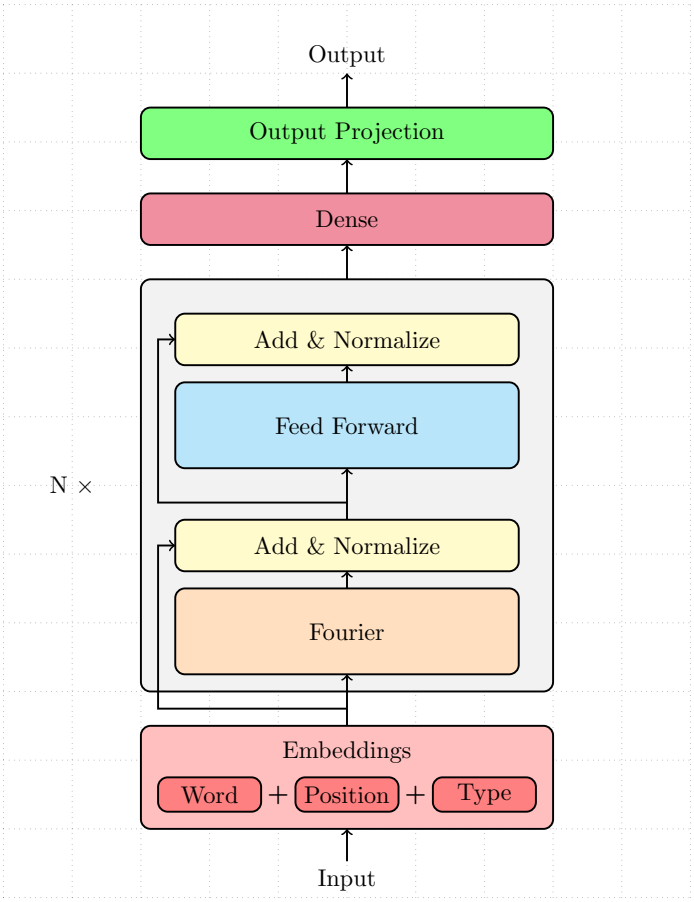


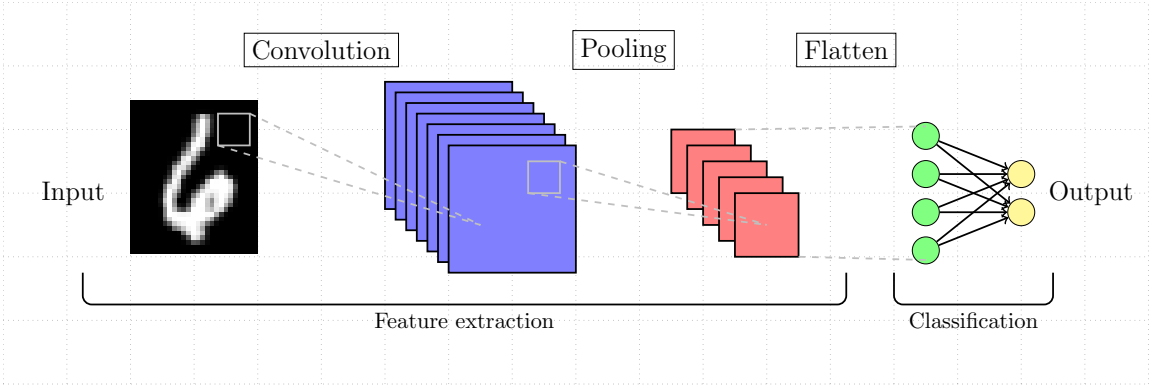
4.3 Turn a 1-star rating into a 5-star rating.

Produce the following image using TikZ and the image file `Star_rating.png`.
Hint: You need to pass some options to `\includegraphics` *inside multiple* `\node`*'s.*

4.4 Drawing a flowchart

Draw the following flowchart using TikZ.
Hint: Objects written later will be formatted on top of objects written earlier.



4.5 Drawing a convolutional neural network

Produce the following diagram using TikZ and the image file `MNIST_sample.png`. *Hint: Use some loops and relative coordinates to draw the convolutional and pooling layers, that way you can move the collection of filters by just changing one set of coordinates in the TikZ code.*