

# APSC 258: Lab 4 Manual

Andre Cox

July 21, 2022

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Understanding Dense Layers . . . . .	2
1.2	Gradient . . . . .	2
1.3	Data . . . . .	2
<b>2</b>	<b>Dense Layers Deep Dive</b>	<b>3</b>
2.1	How dense layers work in practice . . . . .	3
2.2	Using Dense Layers in Keras . . . . .	3
<b>3</b>	<b>Start of the Lab</b>	<b>5</b>
3.1	Data Upload . . . . .	5
<b>4</b>	<b>Running the Code</b>	<b>5</b>
4.1	Building your network . . . . .	5
4.2	Saving the network . . . . .	6
<b>5</b>	<b>Running the Pi Car V</b>	<b>6</b>
5.1	Control the Pi Car V . . . . .	6
5.2	Finishing Up . . . . .	6

# Chapter 1

## Introduction

In this lab we will be creating a simple neural network to predict the steering angle of the car. By the end of this lab you should have a network that can drive the car and follow the lane lines. Some parts about making the neural network are out of the scope of this course so we will provide you with a starting point for your work. You can download the code for this lab from the [Github Repository](#).

### 1.1 Understanding Dense Layers

Dense layers are the most basic type of neural network layer. They are simply a layer of neurons that are connected to each other. The neurons in a dense layer are simply a weighted sum of the inputs. The weights are adjusted until the network is able to predict the correct output. The process of updating the weights is called backpropagation. Generally the more neurons and layers you have the more accurate the network will be.

### 1.2 Gradient

We can train our neural network using our computer. However this will probably be slow depending on how powerful your computer is. To solve this we will use a service called [Gradient](#). This service provides a cloud based platform for training neural networks and it is free for basic use. You can create a free account at the link above and you will be able to train your own neural network.

### 1.3 Data

We will be using 2 datasets. The first is a training dataset that we will use to train our neural network. The second is a test dataset that we will use to test our neural network. You can download the datasets from the link below. These files are compressed using 7zip. Our provided code will extract the files for you. We will show you how to do this in the next section.

**REMEMBER TO PUT THE URL HERE**

## Chapter 2

# Dense Layers Deep Dive

### 2.1 How dense layers work in practice

Dense layers are a array of weights that are placed side by side. Multiple layers of dense layers can be connected together by connecting each weight in the previous layer to the weight in the next layer. This may look something like this:

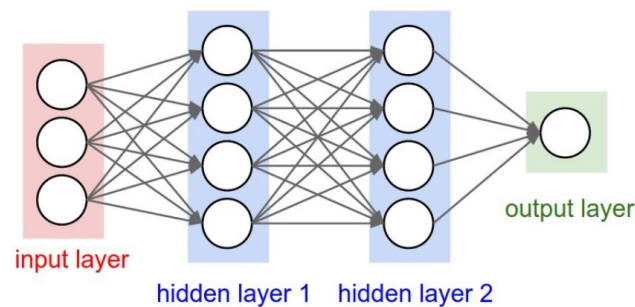


Figure 2.1: Dense layers visualization

The weights in each neuron in the dense layer are activated by the inputs. The activation of the neuron is the weighted sum of the inputs. The activation of the neuron is then passed through a non-linear activation function to get the output. The output of the neuron is then connected to the next layer. There are many different activation functions that can be used. A couple of common ones are displayed below.

### 2.2 Using Dense Layers in Keras

Keras is the Python library that we will use to create our neural network. Keras is a high level library that allows us to create neural networks in a simple and easy way. Later you will see that we provide a pre made template for you to use. However you still need to understand how to create a neural network in Keras.

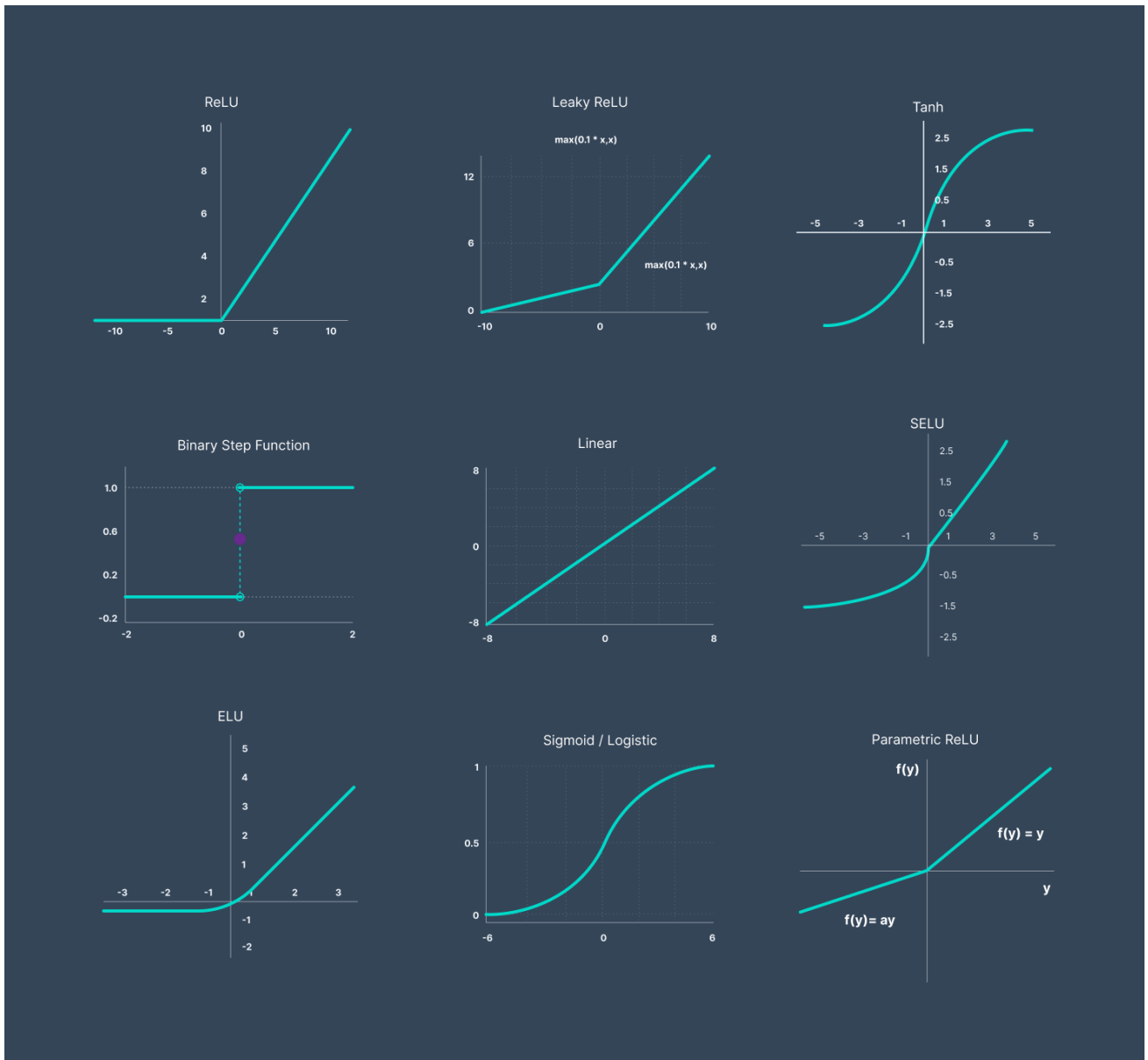


Figure 2.2: Common activation functions

## Chapter 3

# Start of the Lab

First we will download the demo code from the [Github Repository](#). And then extract the files to a folder. Next go to the folder called.

Lab Code/Lab Part 4/

Here you will find a file called model.ipynb. This is the file that we will use to train our neural network. To run this file we will upload it to Gradient and then run it. To do this go to the Gradient website and sign up. Then click "CREATE A PROJECT". You will be asked to choose a name for your project. Pick whatever you want. Then click "CREATE". Now press the "CREATE" button in the middle of the page. When it asks you to select a runtime choose "TensorFlow". Next you can change the auto shutdown to 6 hours which should be more than enough time to train your neural network. Then click "START NOTEBOOK". Once the notebook has started you can go to the left hand side of the page and click the files icon. Then click the upload files icon. Now select the file model.ipynb and click upload. You can then open the uploaded file.

### 3.1 Data Upload

Now we will upload the training and test datasets to Gradient. Simply upload the data.7z file to the project the same way you uploaded the model.ipynb file.

## Chapter 4

# Running the Code

Now that Gradient is set up and your data is uploaded you can run the code. Simply open the model.ipynb file and then run each code block.

### 4.1 Building your network

Once you get to Part 4 of the code you are ready to build your neural network. You can modify the code provided to try to get a better training result by lowering the mean squared error. To do this you can try adding more layers, changing the activation functions, or changing the number of neurons in the hidden layers.

## 4.2 Saving the network

After you have trained your neural network you can save it to a file. To do this go to the final code block and run it. However before you do this we recommend changing the name of the file to something more meaningful. After this block runs you can refresh the file browser in the right hand side of the screen and you will see a file called model.h5 or whatever you decided to name it. This is the file that the Pi Car V will use to self drive. You can download the file by right clicking on it and selecting Download.

# Chapter 5

## Running the Pi Car V

Now that you have trained your neural network you can run the Pi Car V. To do this you can open the Demos folder you downloaded in the first part of this lab and go to the folder "Lab Code/Lab Part 4/". In this folder you will find a file called runNetwork.py. This file will run the Pi Car V. All you need to do is connect to the Pi Car V's wifi network. Then put the model.h5 file in the same folder as the runNetwork.py file. Then run the file by typing "python runNetwork.py". If your file is named something else you can add some command line arguments to the file. For example "python runNetwork.py -model differentModel.h5". For a full list of command line arguments you can type "python runNetwork.py -help".

### 5.1 Control the Pi Car V

Once the runNetwork file is running a window will open. You can use this window to adjust the speed of the car and the color to mask.

### 5.2 Finishing Up

To complete this lab you will need to experiment to get your car to follow the lane lines. You may find that this is a bit tricky. Next lab we will introduce a new layer that will help you with this.