1495 UNIVERSITY OF ABERDEEN | CELEBRATING 525 YEARS 1495 – 2020

ABERDEEN 2040

Train a Deep Feed-Forward Neural Network Demonstration

Did it Rain in Seattle from 1948 to 2017?

CSV Dataset

Coding, GIS and Remote Sensing Data for Glaciology

24 April 2024

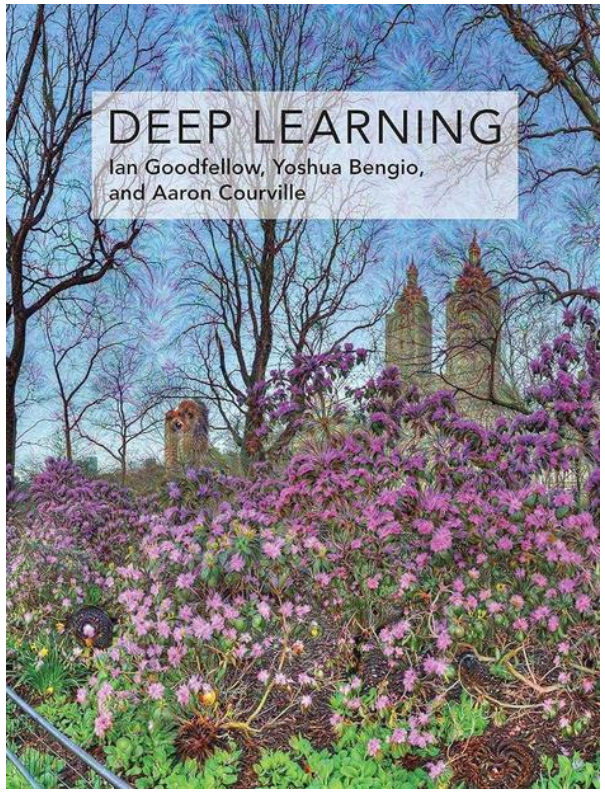
Meston G05

Overview

1. Background and Theory
2. Training
3. Evaluation
4. Code Run Though

https://cs231n.stanford.edu/

Additional Reading



Deep Learning – Ian Goodfellow, Yoshua Bengio and Aaron Courville 2017

Python Code for the Session

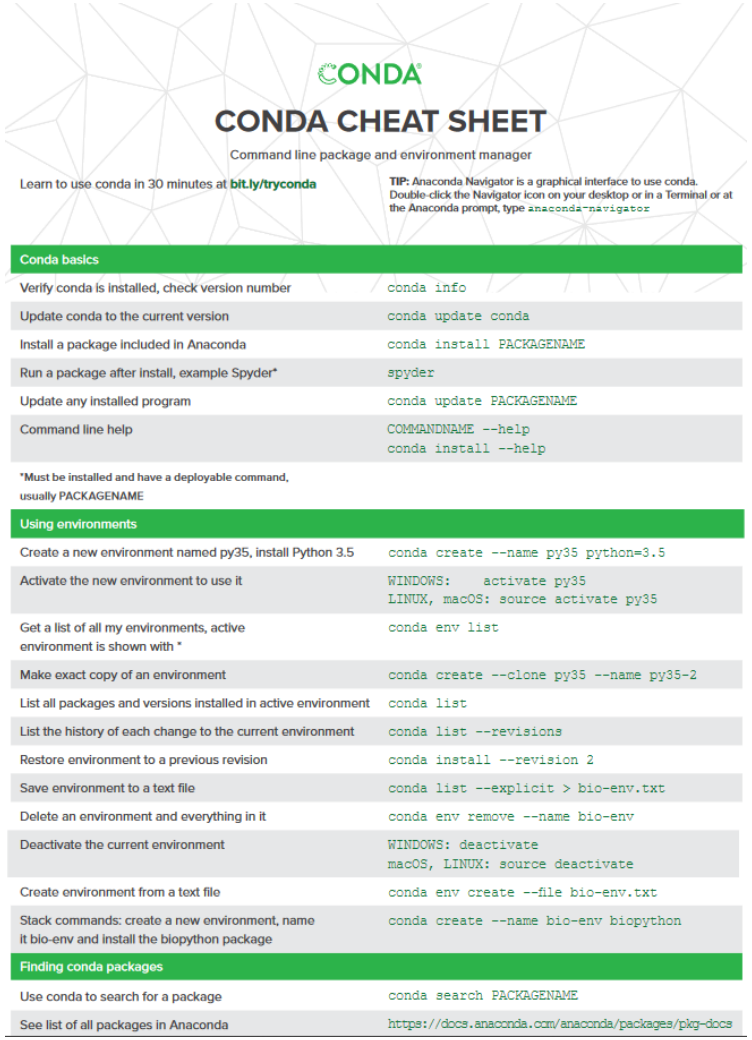
The code for the session can be found implemented in PyTorch and TensorFlow at:

GitHub ID: Stevieee83

https://github.com/Stevieee83

For our session we will use the PyTorch version but please feel free to use the TensorFlow version if you wish.

Anaconda Cheat Sheet



CONDA

CONDA CHEAT SHEET

Command line package and environment manager

Learn to use conda in 30 minutes at **bit.ly/tryconda**

TIP: Anaconda Navigator is a graphical interface to use conda. Double-click the Navigator icon on your desktop or in a Terminal or at the Anaconda prompt, type `anaconda-navigator`

Conda basics	
Verify conda is installed, check version number	`conda info`
Update conda to the current version	`conda update conda`
Install a package included in Anaconda	`conda install PACKAGENAME`
Run a package after install, example Spyder*	`spyder`
Update any installed program	`conda update PACKAGENAME`
Command line help	`COMMANDNAME --help` `conda install --help`

*Must be installed and have a deployable command, usually PACKAGENAME

Using environments	
Create a new environment named py35, install Python 3.5	`conda create --name py35 python=3.5`
Activate the new environment to use it	`WINDOWS: activate py35` `LINUX, macOS: source activate py35`
Get a list of all my environments, active environment is shown with *	`conda env list`
Make exact copy of an environment	`conda create --clone py35 --name py35-2`
List all packages and versions installed in active environment	`conda list`
List the history of each change to the current environment	`conda list --revisions`
Restore environment to a previous revision	`conda install --revision 2`
Save environment to a text file	`conda list --explicit > bio-env.txt`
Delete an environment and everything in it	`conda env remove --name bio-env`
Deactivate the current environment	`WINDOWS: deactivate` `macOS, LINUX: source deactivate`
Create environment from a text file	`conda env create --file bio-env.txt`
Stack commands: create a new environment, name it bio-env and install the biopython package	`conda create --name bio-env biopython`

Finding conda packages	
Use conda to search for a package	`conda search PACKAGENAME`
See list of all packages in Anaconda	`https://docs.anaconda.com/anaconda/packages/pkg-docs`

Installing and updating packages	
Install a new package (Jupyter Notebook) in the active environment	`conda install jupyter`
Run an installed package (Jupyter Notebook)	`jupyter-notebook`
Install a new package (toolz) in a different environment (bio-env)	`conda install --name bio-env toolz`
Update a package in the current environment	`conda update scikit-learn`
Install a package (boltons) from a specific channel (conda-forge)	`conda install --channel conda-forge boltons`
Install a package directly from PyPI into the current active environment using pip	`pip install boltons`
Remove one or more packages (toolz, boltons) from a specific environment (bio-env)	`conda remove --name bio-env toolz boltons`

Managing multiple versions of Python	
Install different version of Python in a new environment named py34	`conda create --name py34 python=3.4`
Switch to the new environment that has a different version of Python	`Windows: activate py34` `Linux, macOS: source activate py34`
Show the locations of all versions of Python that are currently in the path **NOTE:** The first version of Python in the list will be executed.	`Windows: where python` `Linux, macOS: which -a python`
Show version information for the current active Python	`python --version`

Specifying version numbers

Ways to specify a package version number for use with conda create or conda install commands, and in meta.yaml files.

Constraint type	Specification	Result
Fuzzy	`numpy=1.11`	1.11.0, 1.11.1, 1.11.2, 1.11.18 etc.
Exact	`numpy==1.11`	1.11.0
Greater than or equal to	`"numpy>=1.11"`	1.11.0 or higher
OR	`"numpy=1.11.1|1.11.3"`	1.11.1, 1.11.3
AND	`"numpy>=1.8,<2"`	1.8, 1.9, not 2.0

NOTE: Quotation marks must be used when your specification contains a space or any of these characters: > < | *

MORE RESOURCES

Free Community Support	groups.google.com/a/continuum.io/forum/#!forum/conda
Online Documentation	conda.io/docs
Command Reference	conda.io/docs/commands
Paid Support Options	anaconda.com/support
Anaconda Onsite Training Courses	anaconda.com/training
Anaconda Consulting Services	anaconda.com/consulting

Follow us on Twitter @anacondainc and join the #AnacondaCrew!

Connect with other talented, like-minded data scientists and developers while

https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf

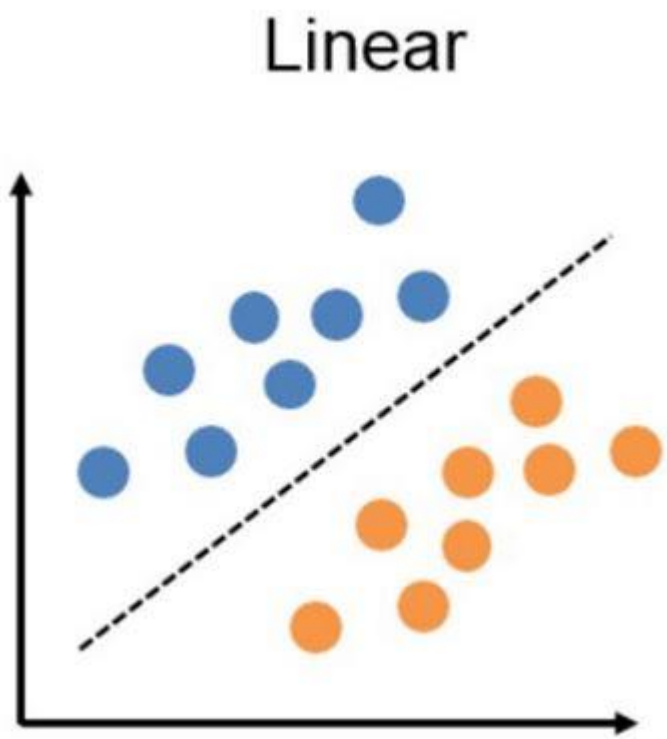
Did it Rain in Seattle from 1948 to 2017?

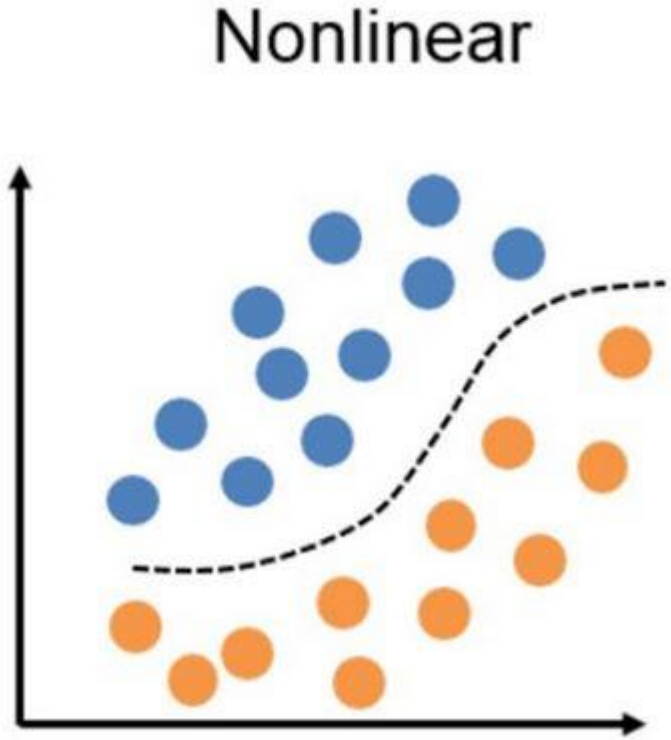
- Tabular CSV dataset with over 25 000 examples

Kaggle.com Dataset Link:

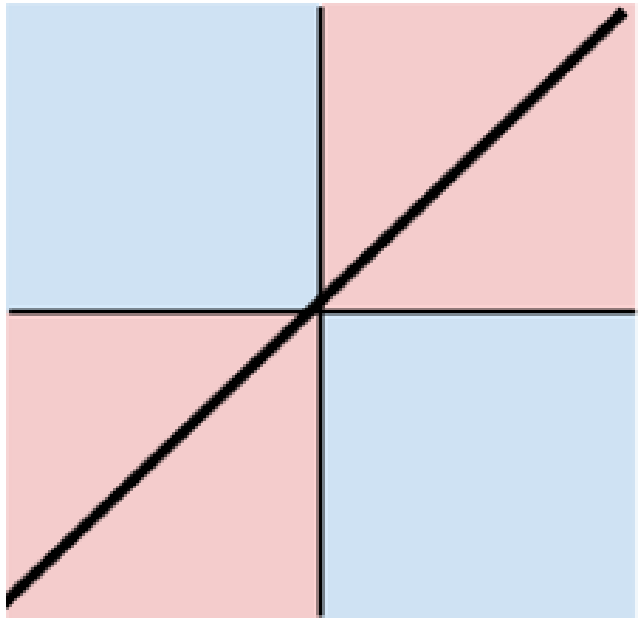
https://www.kaggle.com/datasets/rtatman/did-it-rain-in-seattle-19482017

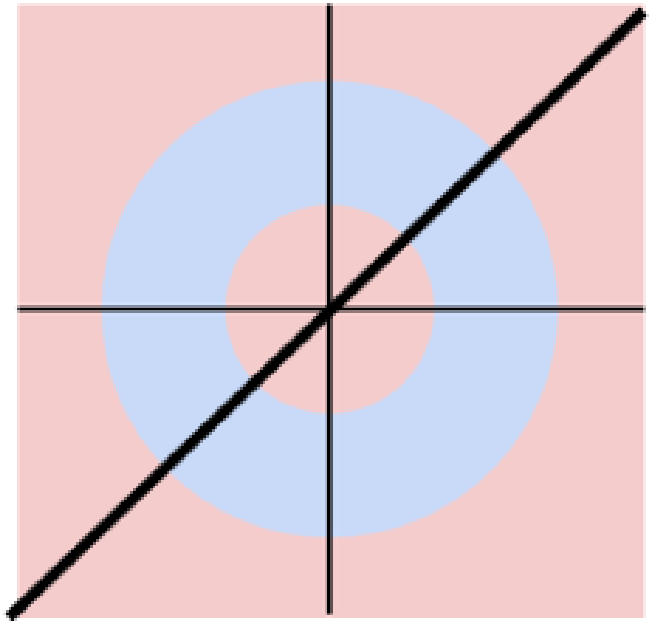
Exploratory Data Analysis

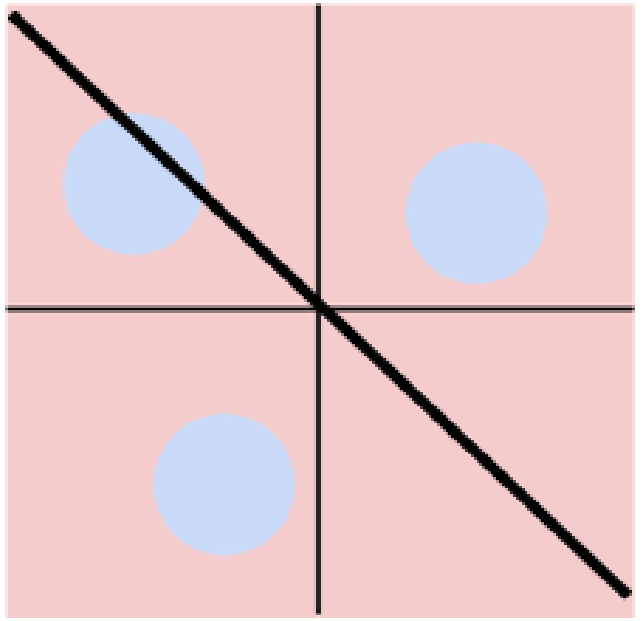




Linear Classifiers

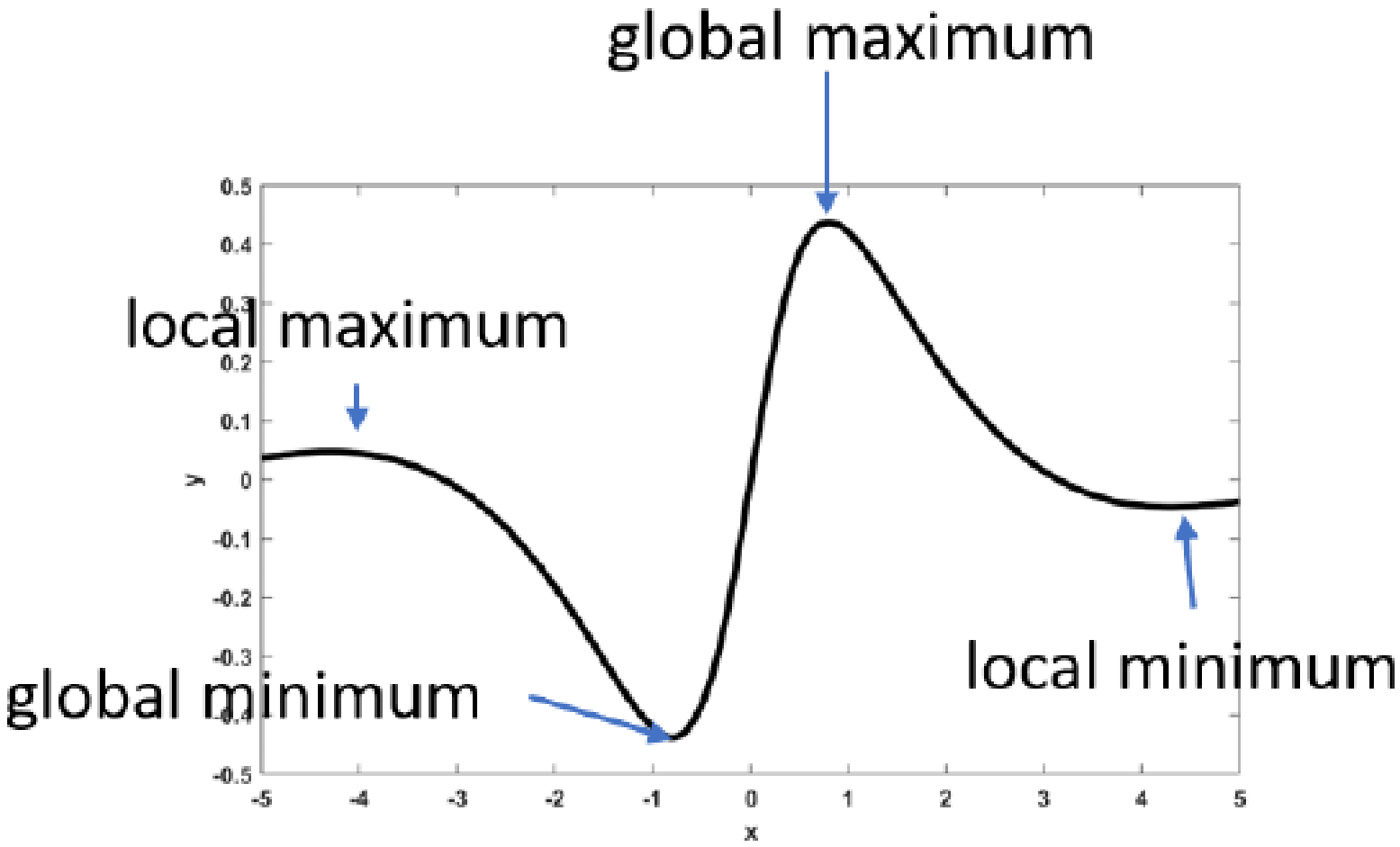




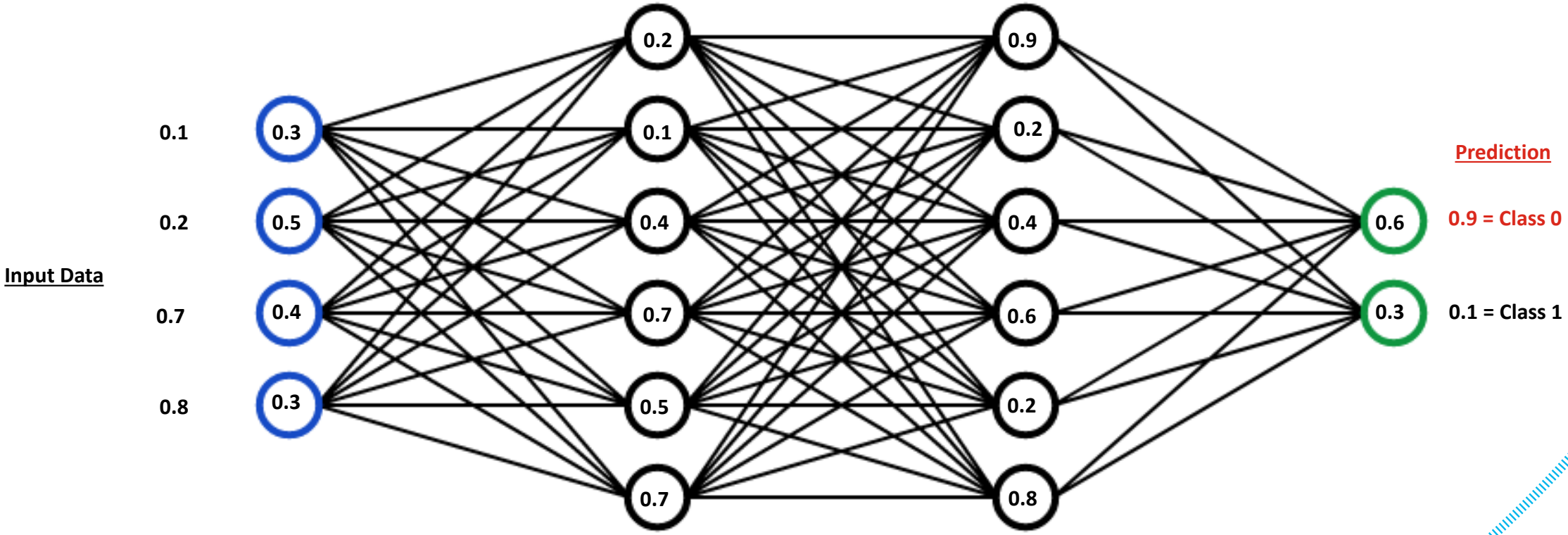


https://cs231n.stanford.edu/

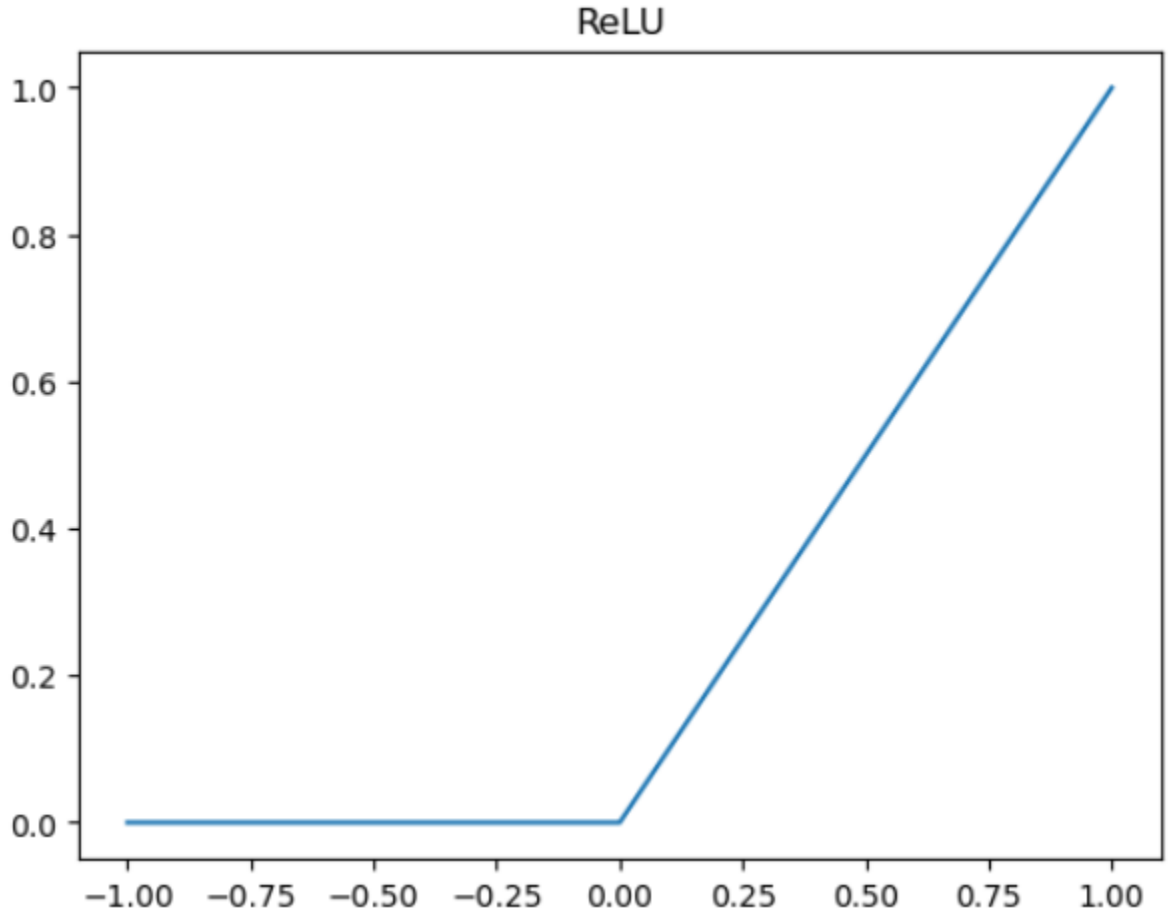
Gradient Descent



Feed-Forward Neural Network (FNN)

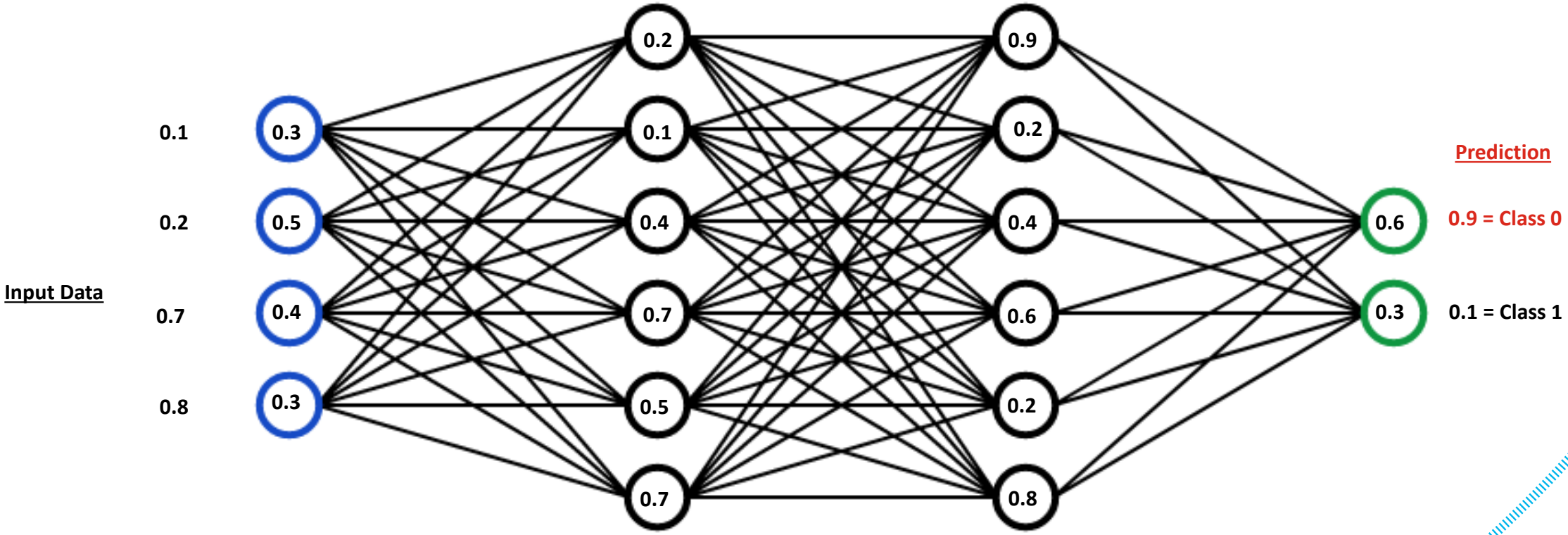


Rectified Linear Unit (ReLU)

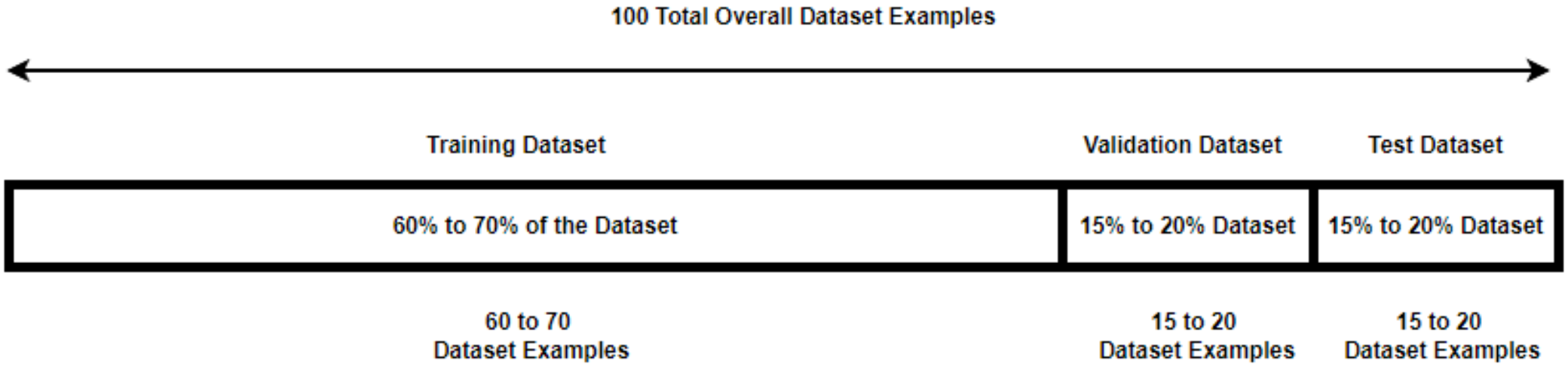


$$f(x) = x^+ = \max(0, x) = \frac{x + |x|}{2} = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \qquad f'(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x < 0. \end{cases}$$

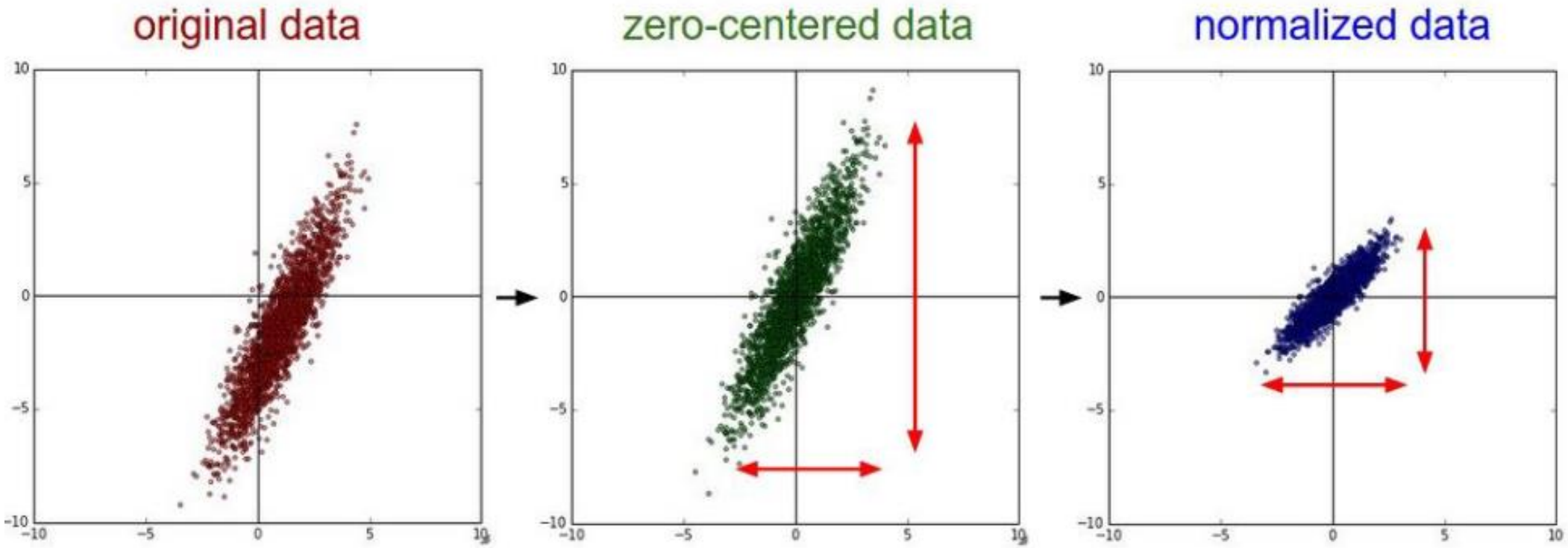
Feed-Forward Neural Network (FNN)



Hold Out Cross Validation



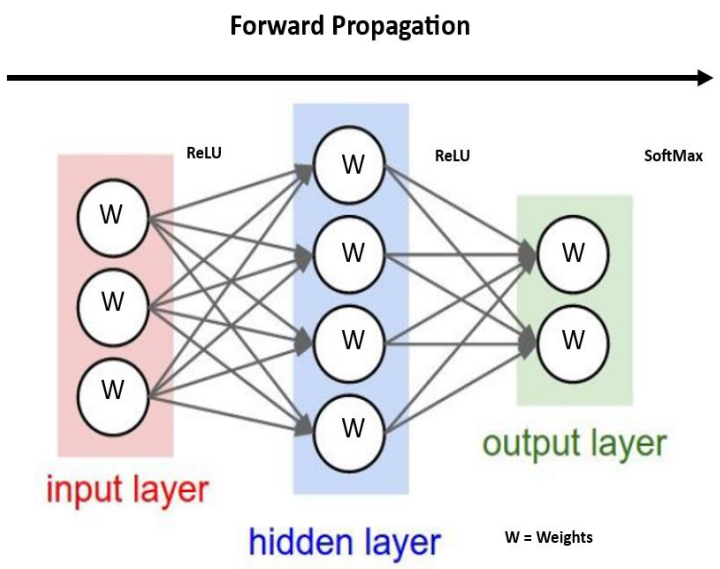
Data Normalisation

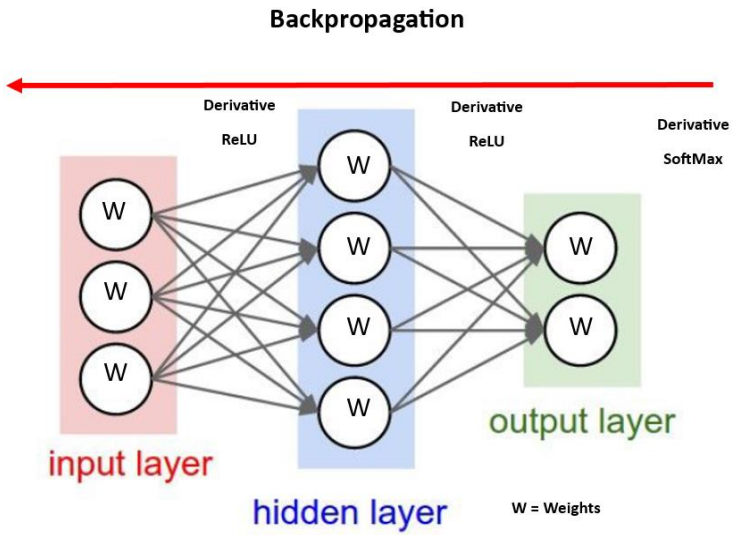


https://cs231n.stanford.edu/

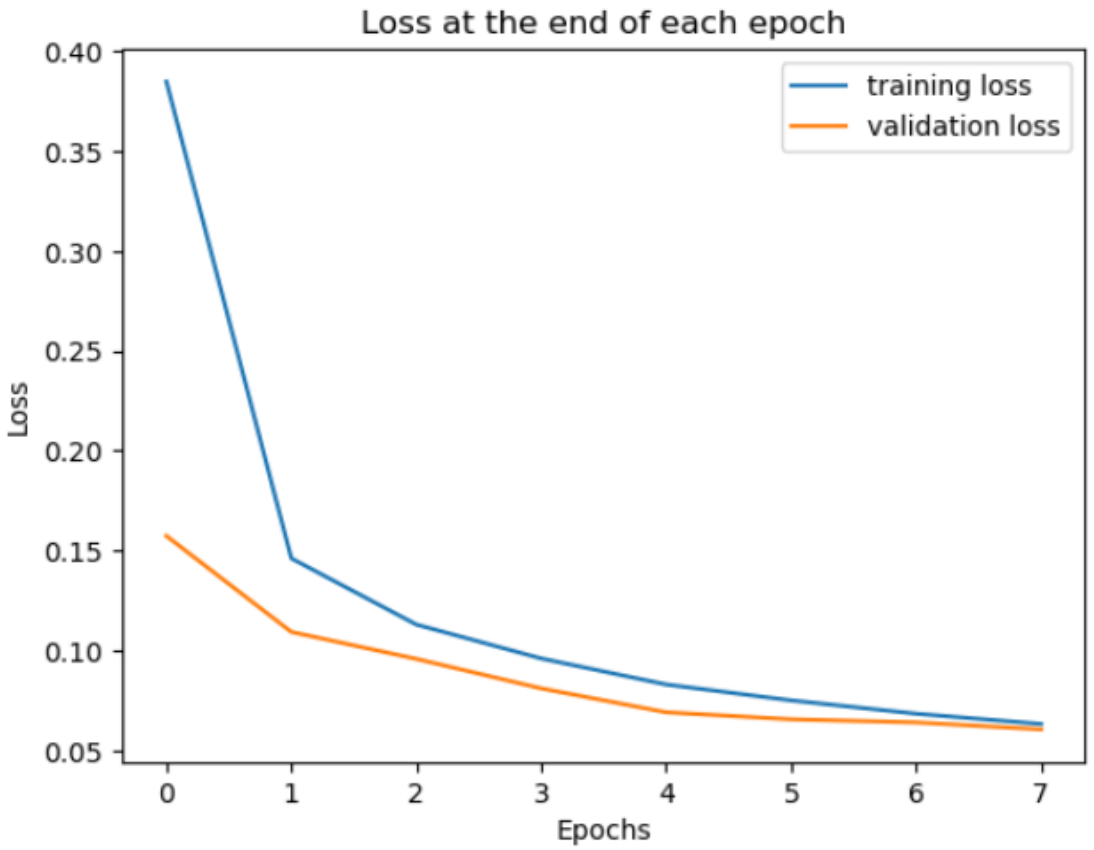
Training the FNN

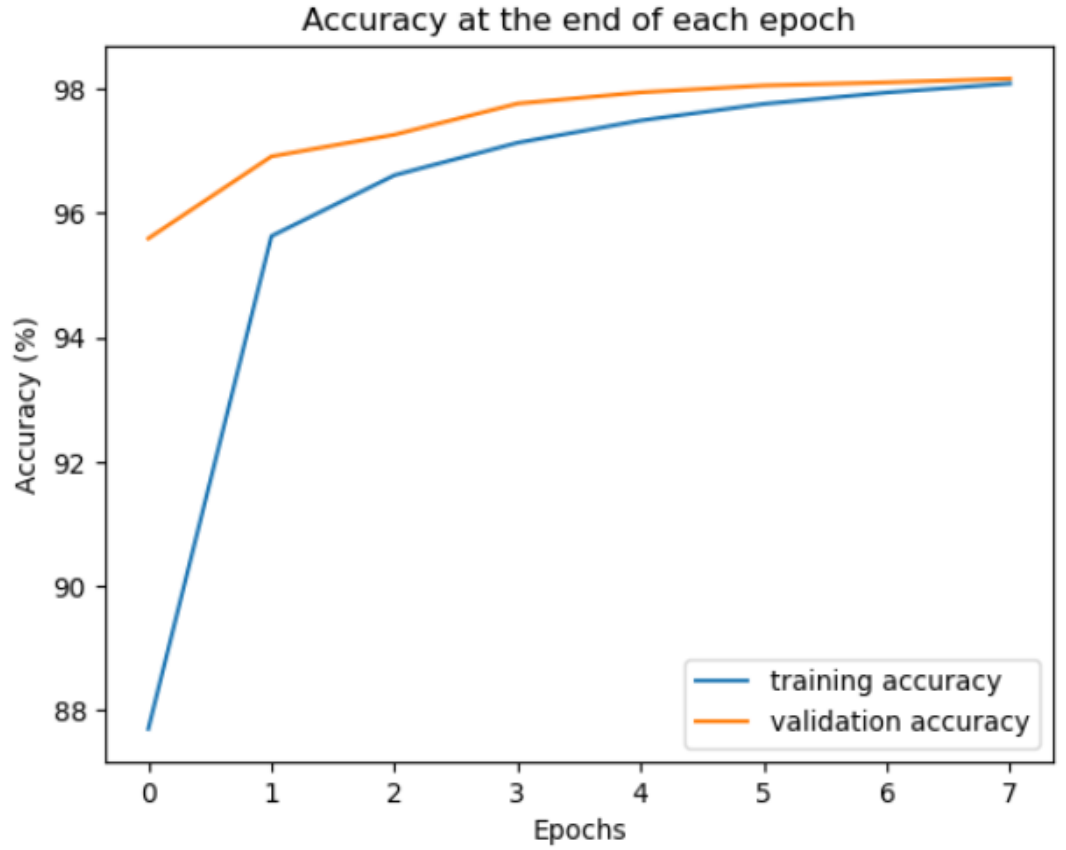
- Set the Learning Rate
- Scale up the Network Parameters
- Adapt the Training Epochs if Required





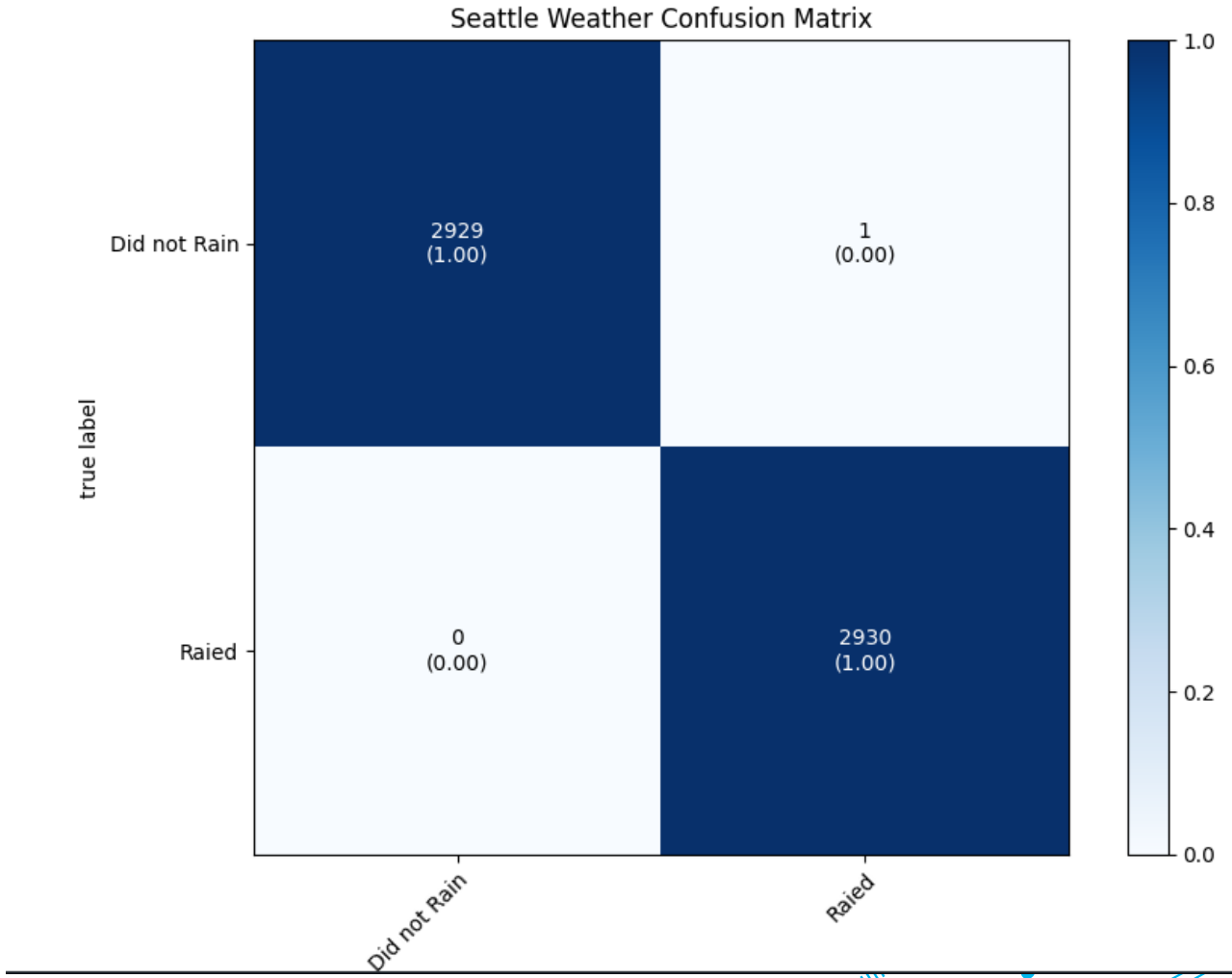
Training the FNN





Evaluating the FNN

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix



Code Run Through

```
# Hyperparameters
input_layer = 4         # Sets the number of parameters in the input layer
hidden_layer_1 = 7      # Sets the number of parameters in hidden layer 1
hidden_layer_2 = 7      # Sets the number of parameters in hidden layer 2
hidden_layer_3 = 7      # Sets the number of parameters in hidden layer 3
hidden_layer_4 = 7      # Sets the number of parameters in hidden layer 4
hidden_layer_5 = 7      # Sets the number of parameters in hidden layer 5
hidden_layer_6 = 7      # Sets the number of parameters in hidden layer 6
output_layer = 1        # Sets the number of parameters in the output layer
lr = 1e-3               # Sets the leanring rate
epochs = 50             # Sets the number of training epochs
batch_size = 32         # Sets the batch size
# -----------------
```

Summary

- Understanding gradient descent and how the learning rate affects the neural network is essential for training deep neural networks.
- Before training the deep neural network, data pre-processing and exploratory data analysis are essential.
- Deep neural networks require large amounts of data but solve problems with non-linear data points, including data with complex statistical trends.

References

Goodfellow, I., Bengio, Y. and Courville, A. (2017) Deep learning. Cambridge, MA: The MIT Press.

Li, F.-F. CS231N: Deep Learning for Computer Vision, Stanford University CS231n: Deep Learning for Computer Vision. Available at: https://cs231n.stanford.edu/ (Accessed: 24 April 2024).

Tatman, R. (2017) Did it rain in Seattle? (1948-2017), Kaggle. Available at: https://www.kaggle.com/datasets/rtatman/did-it-rain-in-seattle-19482017 (Accessed: 01 May 2024).