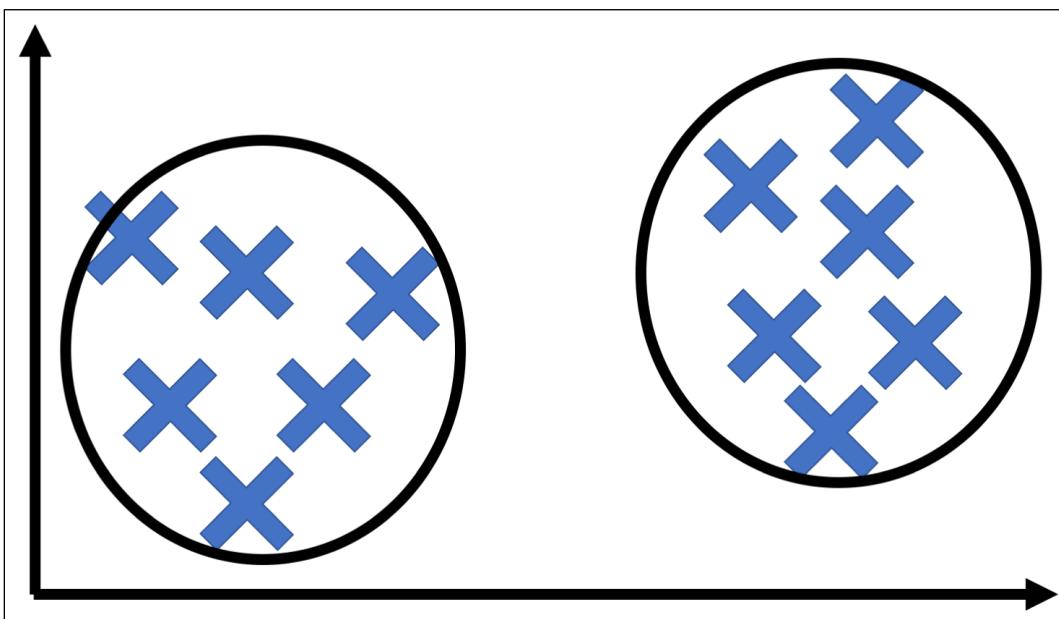
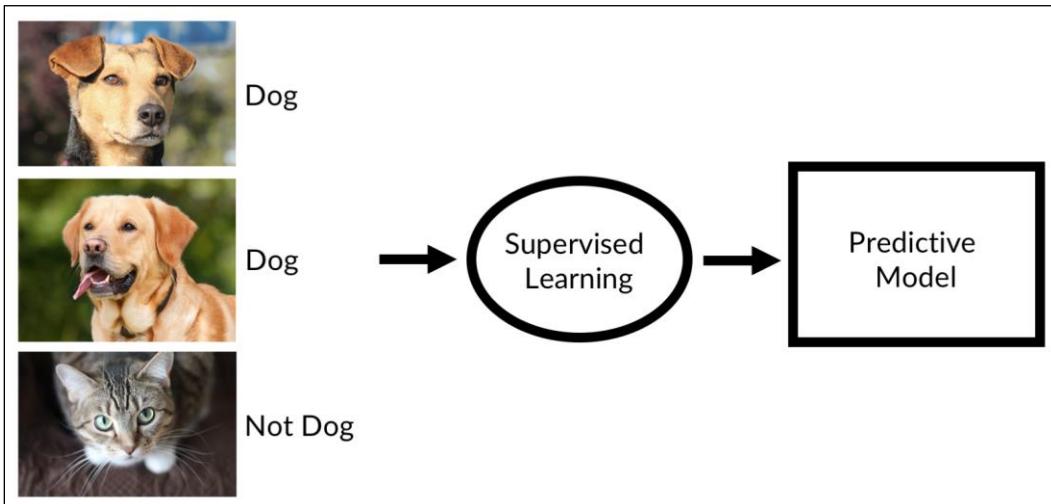
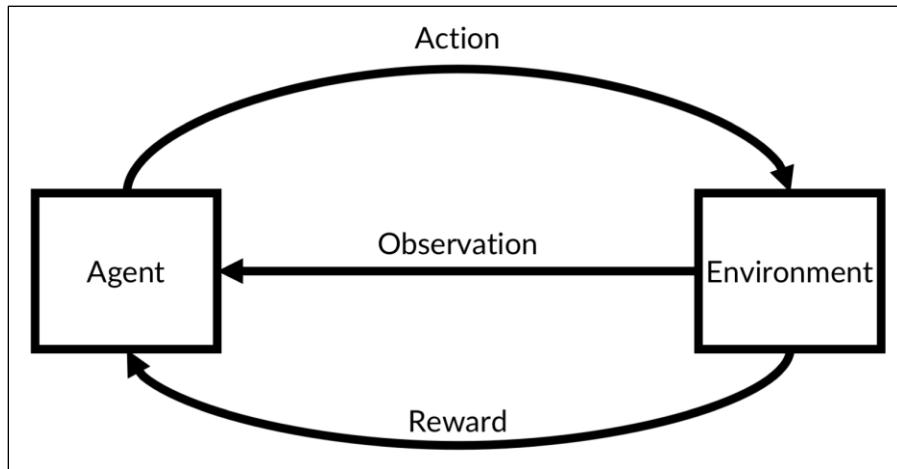


Chapter 1: Neural Networks and Gradient-Based Optimization





Screenshot of the Kaggle Kernels page (<https://www.kaggle.com/kernels>)

The page displays a list of public kernels:

Rank	User	Title	Published	Score	Tags	Actions
328		Titanic: 2nd degree families and majority voting	15m ago in Titanic: Machine Learning from Disaster	0.81818	eda, data visualizati...	View Edit Rmd 181
466		Head Start for Data Scientist	2h ago in Titanic: Machine Learning from Disaster	0.9761	learning, tutorial, beginner	View Edit Rmd 164
12		Kaggle-runnable version of Baris Kanber's LightGBM	5h ago in TalkingData AdTracking Fraud Detection Challenge	0.9761		View Edit Py 4
51		Simpsons World Revealed	3h ago in The Simpsons by the Data	0.9761	animation, tutorial, beginner, nlp, data visuali...	View Edit Rmd 29

K Machine Learning For Finance - Chapter 1 Code Draft saved

File Edit Insert Run Help + Add Dataset ✓ Commit →

Upload Notebook

Machine Learning For Finance - Chapter 1

This Kernel contains the code samples for the chapter 1 of my [Machine Learning For Finance book](#). Note that the original text feature far more text, explanations and figures. This notebook only features the code and some related comments. If you enjoy this content, take a look at the [book](#)

A logistic regressor

The simplest neural network is a logistic regressor. Logistic regression takes in values of any range but outputs only values between zero and one. There are many applications for logistic regressors. One example use case is to predict the likelihood of a homeowner to default on a mortgage. We might take all kinds of values into account to predict the likelihood of default, the debtor's salary, whether she has a car, the security of her job, etc., but the likelihood will always be a value between zero and one. Even the worst debtor ever cannot have a default likelihood above 100% and the best cannot go below 0%.

We will use a library called numpy which enables easy and fast matrix operations in Python. To ensure we get the same result in all of our experiments, we have to set a random seed.

```
In[]: import numpy as np
np.random.seed(1)
```

Markdown Cell

Code Cell

Enable GPU →

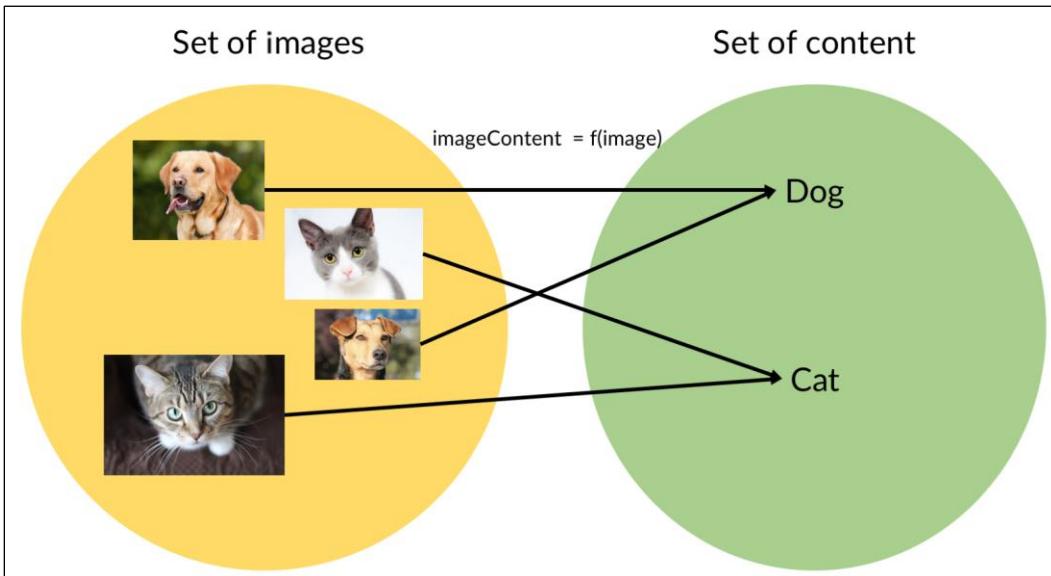
Sessions Draft Session | 3m/9h | GPU Off 0.11%
CPU 237.5MB/16GB
RAM 266.3MB/4.9GB
Disk

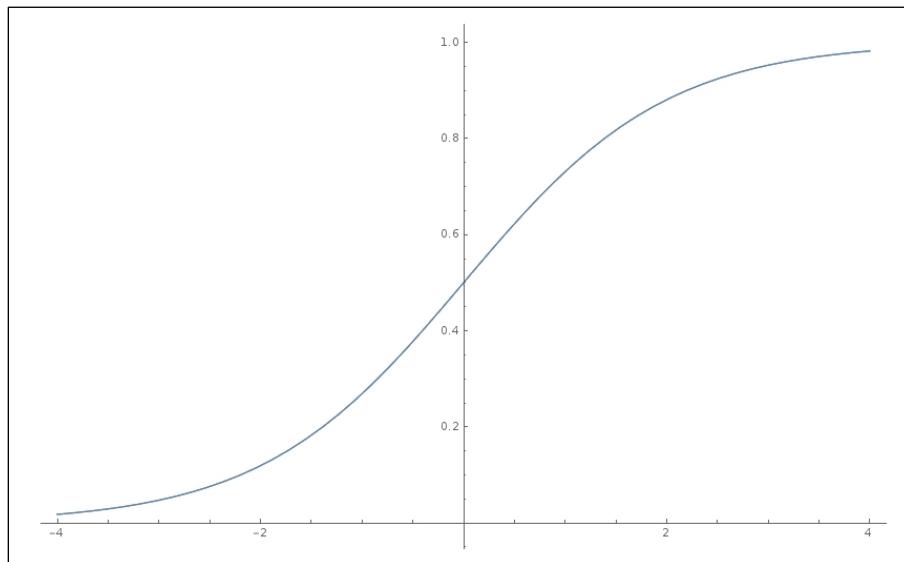
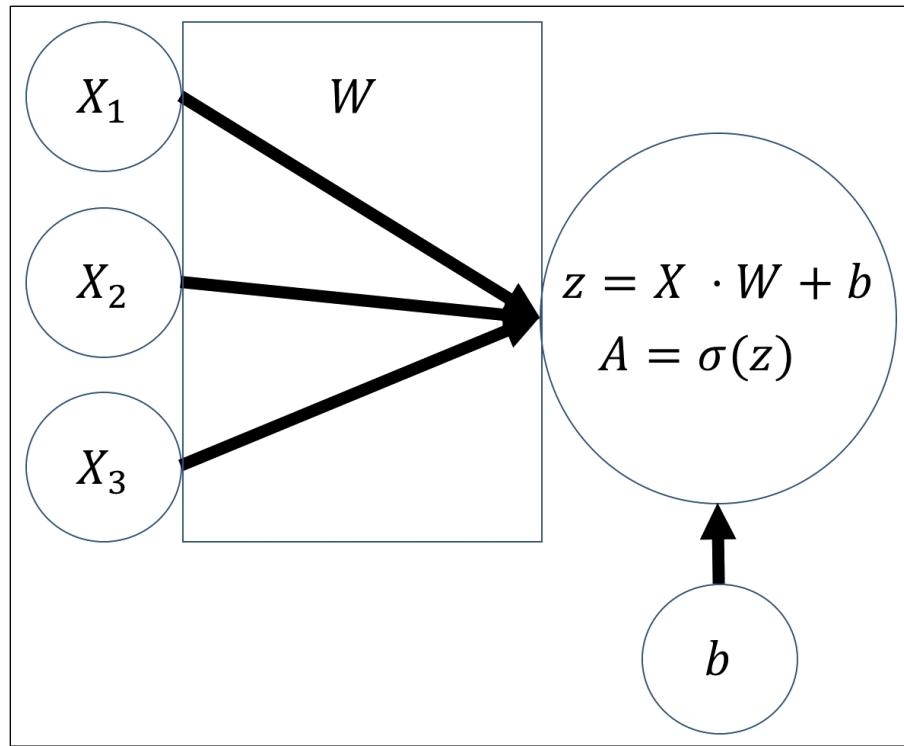
Workspace input (read-only data)

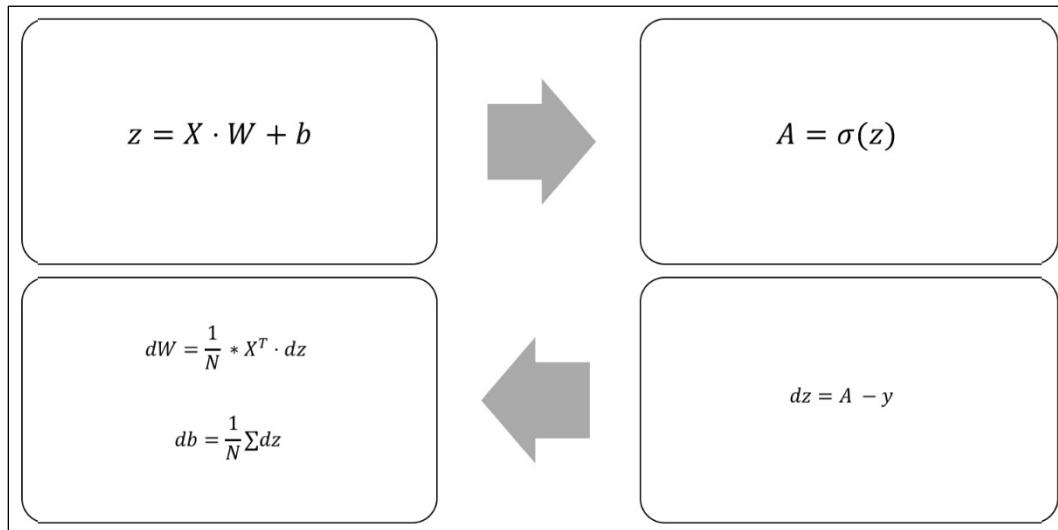
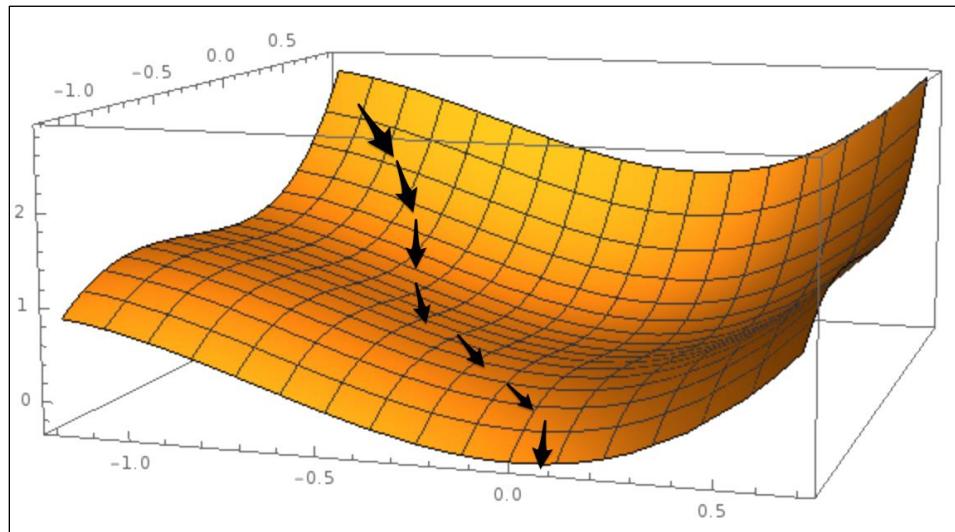
Versions 1 uncommitted draft Jannes Klaas's draft
2 committed versions V2 1y ago +94 -4 V1 1y ago +436 0

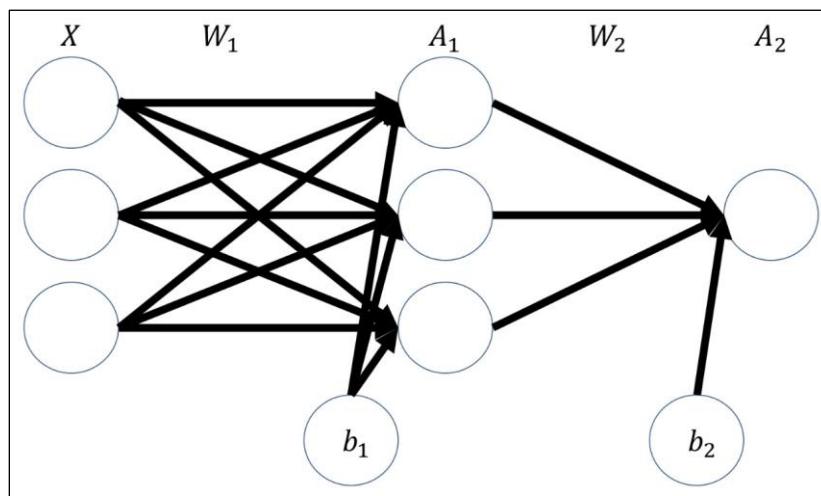
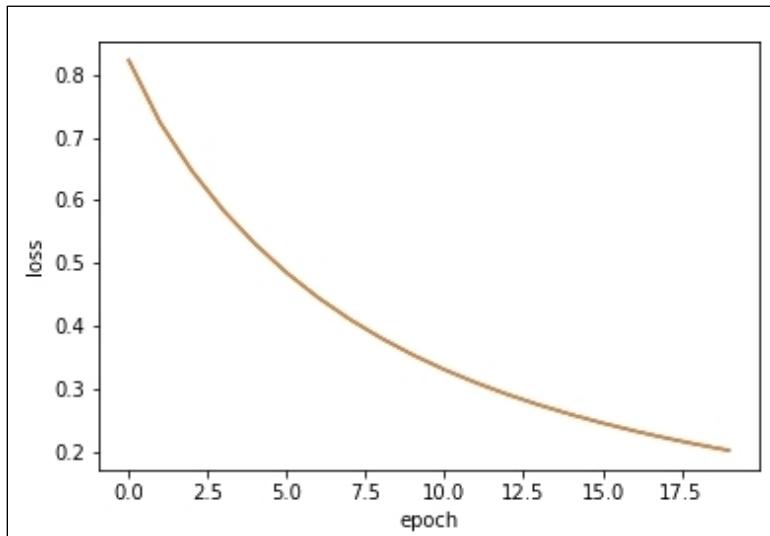
Settings Sharing Public Language Python Docker Latest Available GPU ⚡ Off Internet ⚡ Off

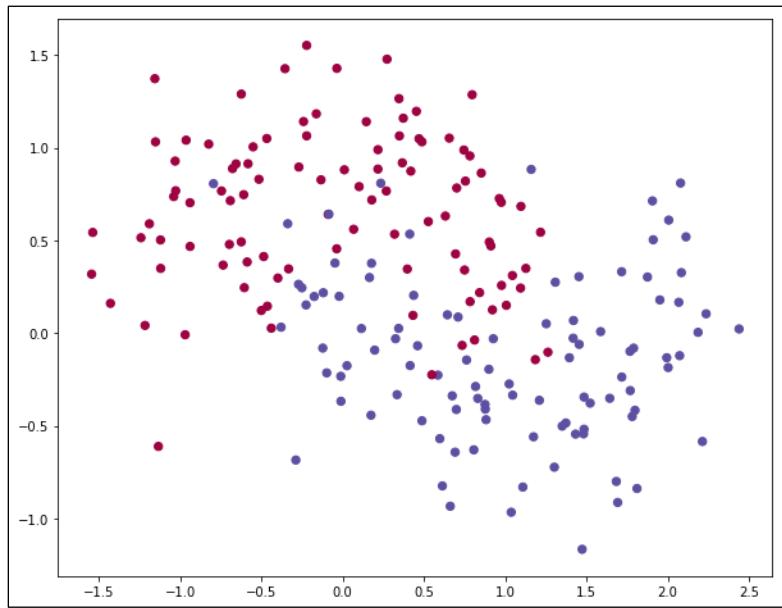
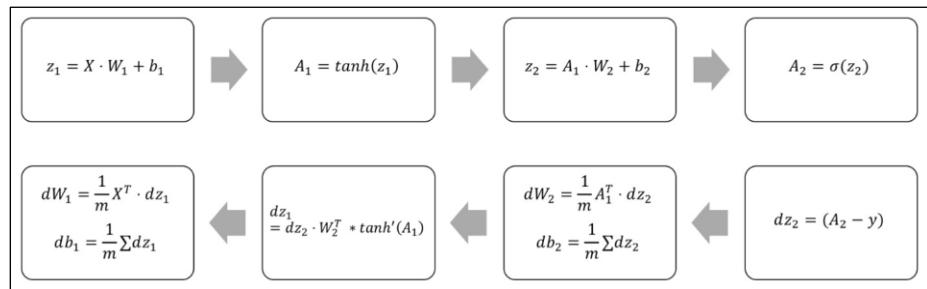
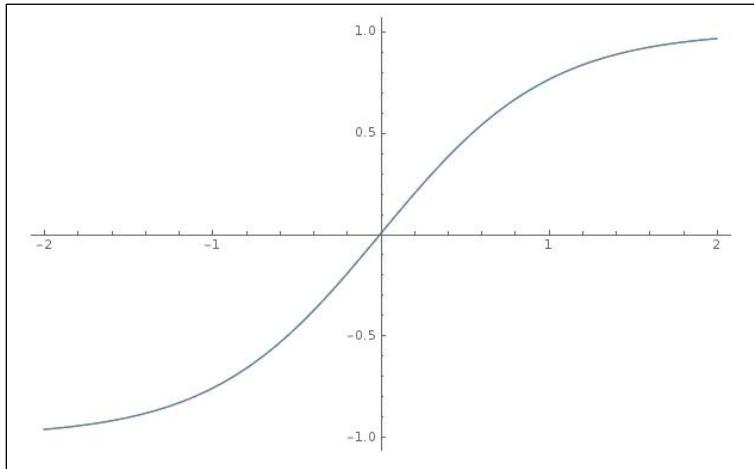
Draft Session | CPU 0% | RAM 237.5MB/16GB | GPU Off | Disk 266.3MB/4.9GB

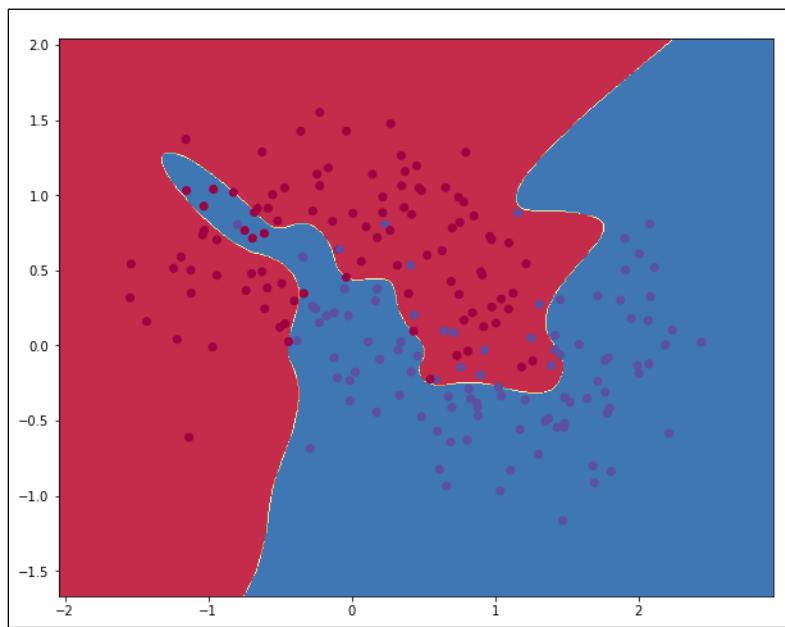
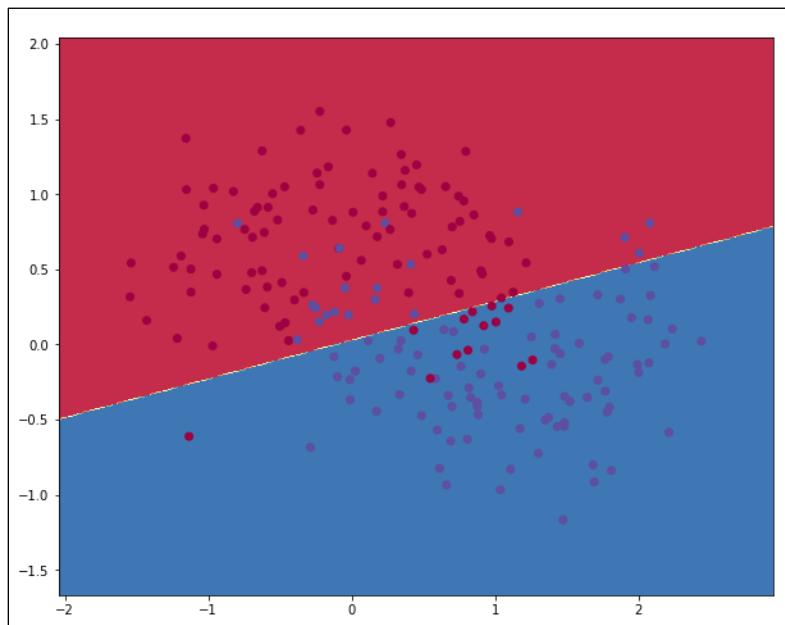


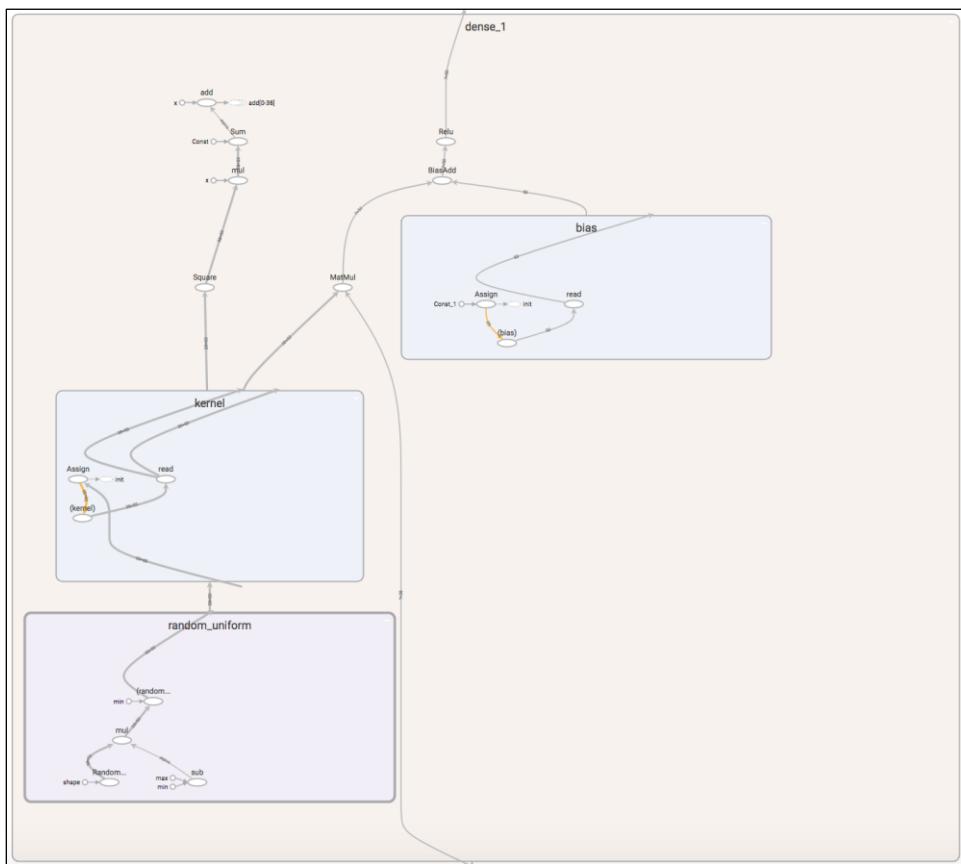
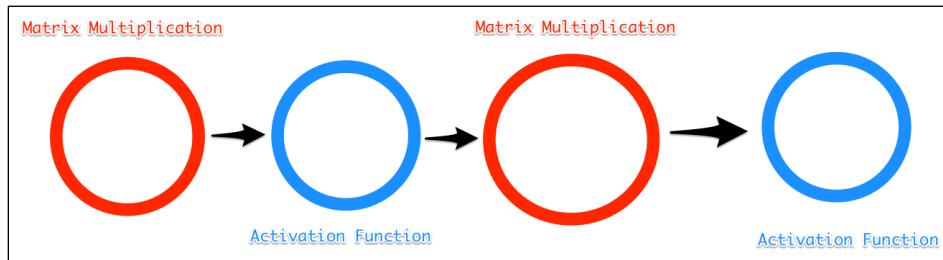




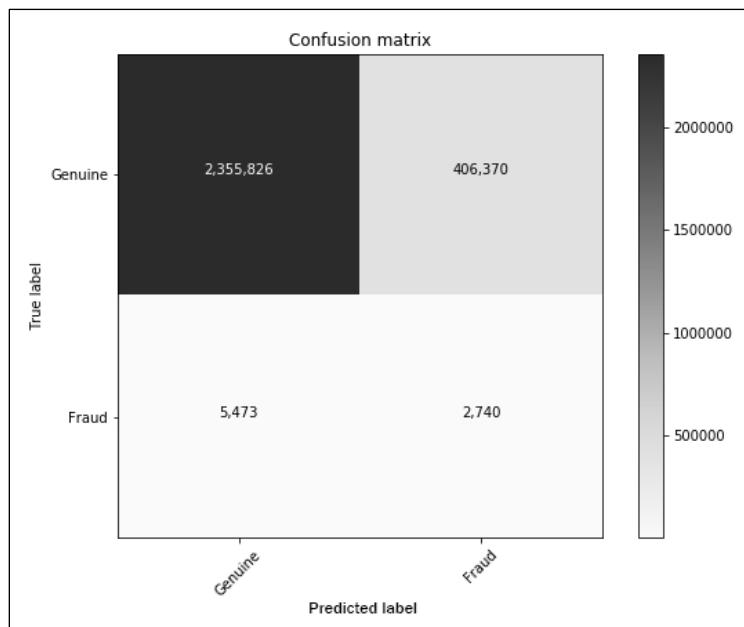
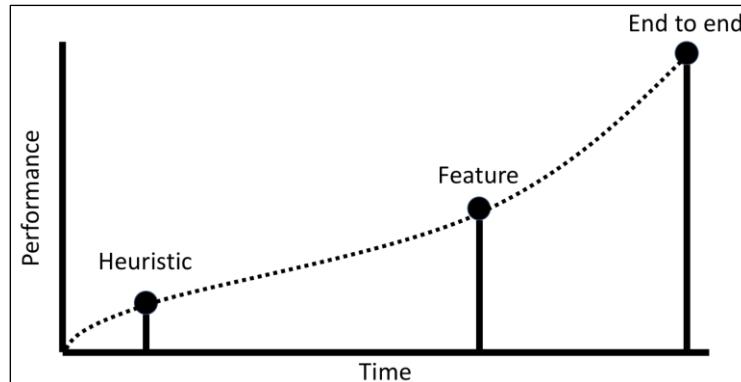


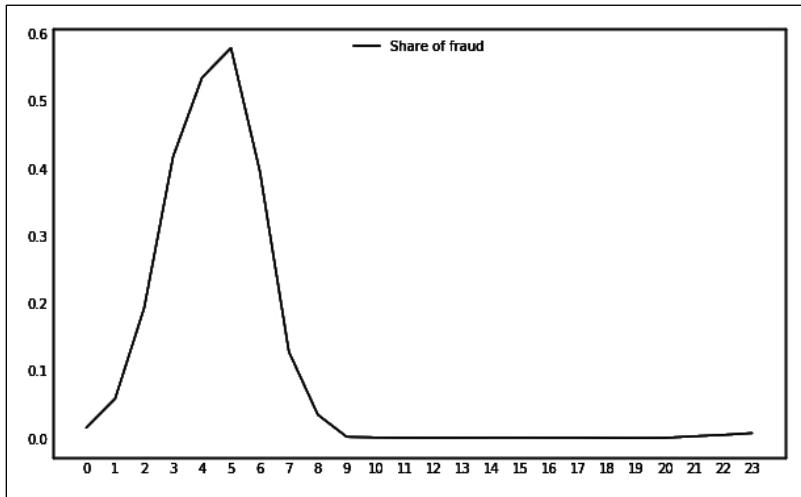
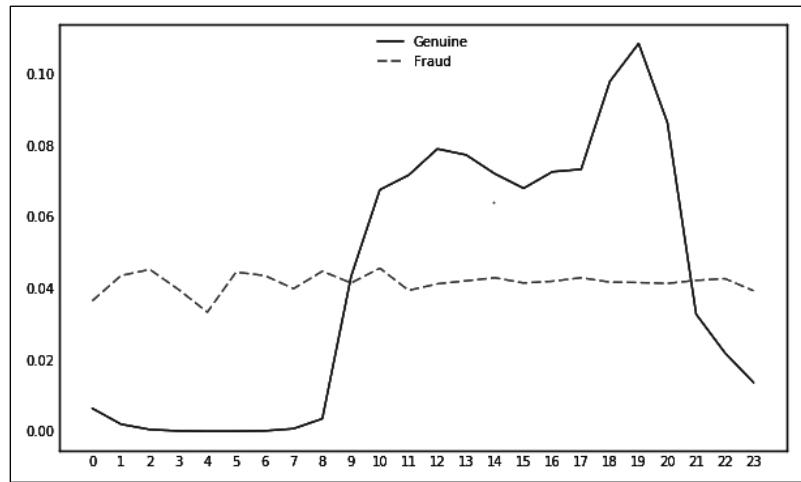


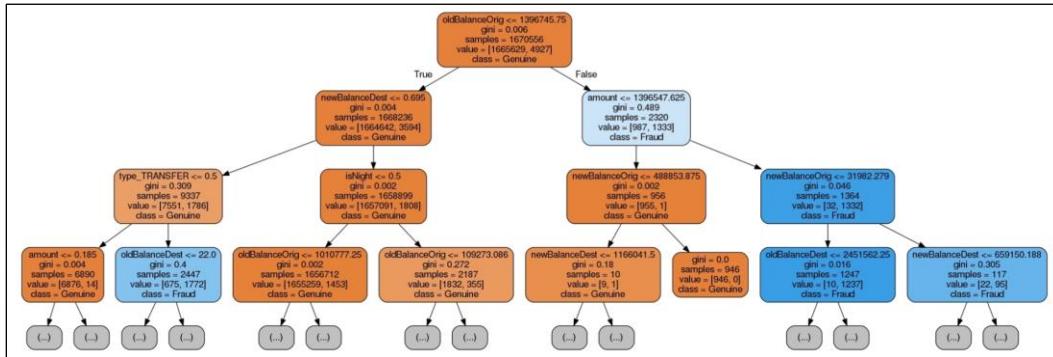
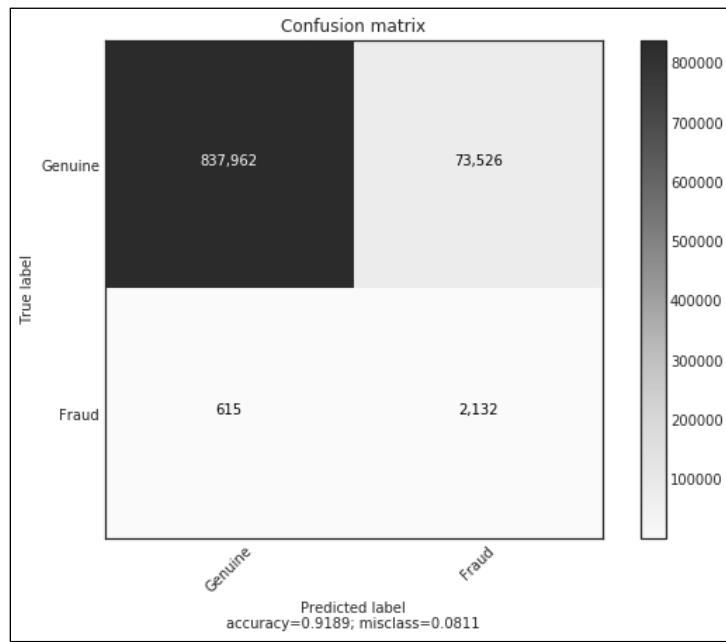


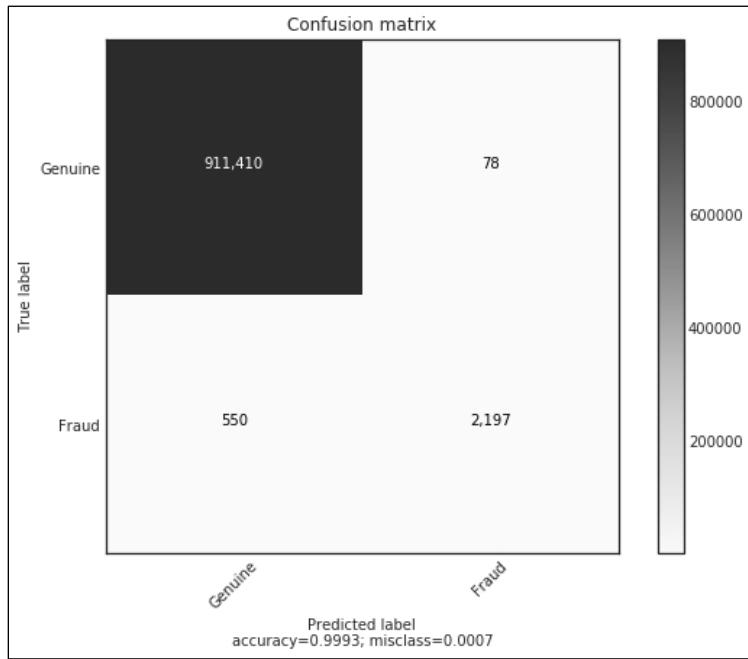


Chapter 2: Applying Machine Learning to Structured Data



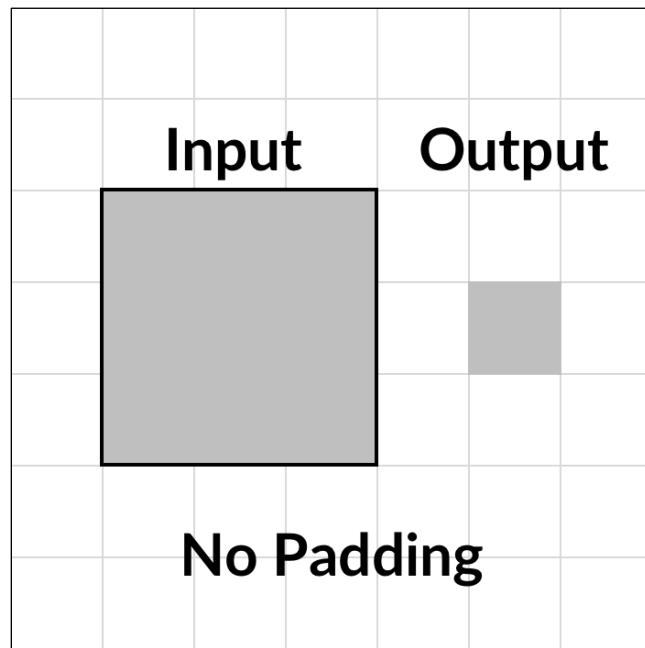
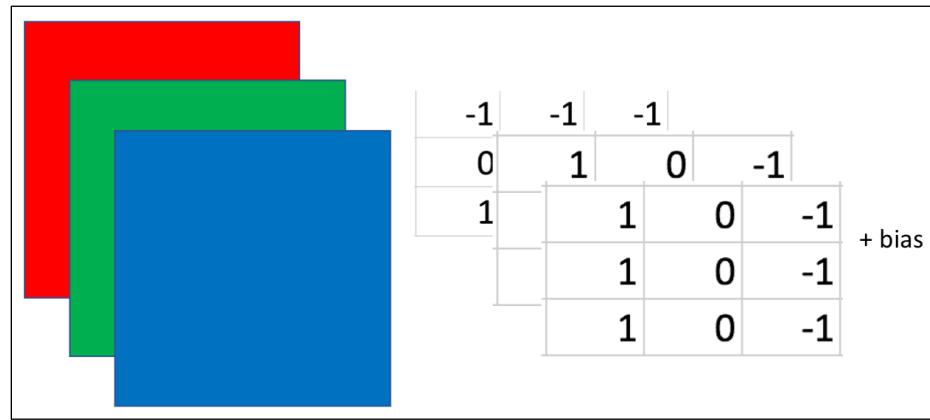


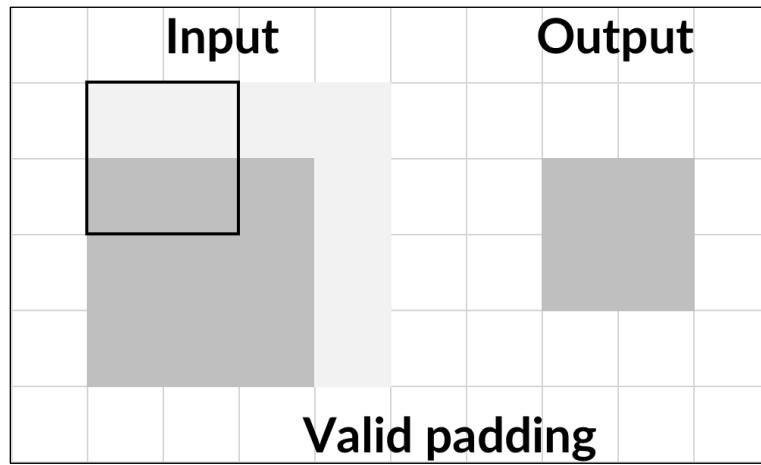
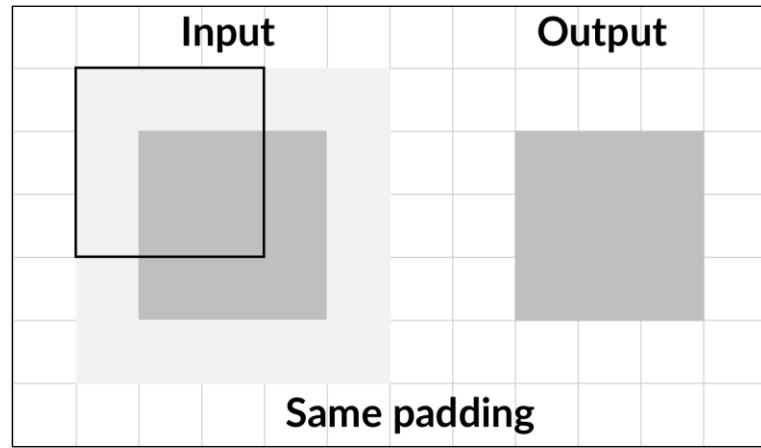


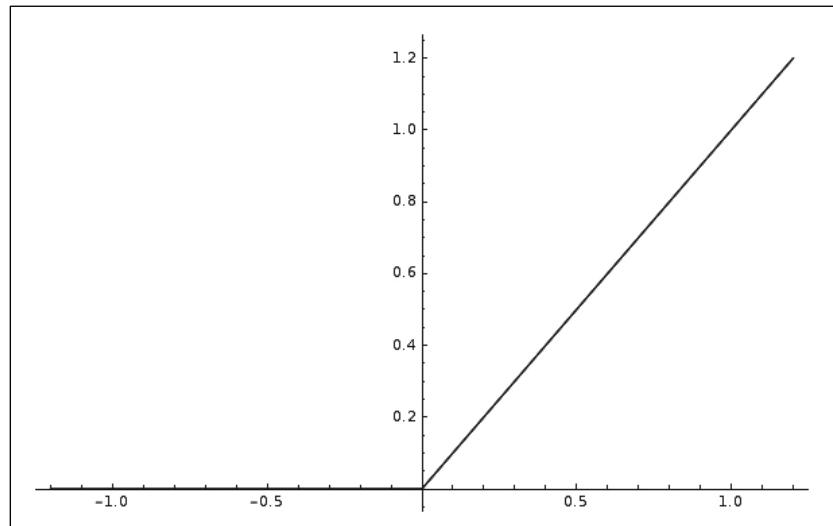


0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-1	-1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
-1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	1	0	1	2	1	0	0	0	0	0	0
-1	-2	-1	-0	0	0	0	1	0	0	0	0	0	-1	-0	1	2	2	1	0	0	0	0	0	0
-2	-2	0	1	0	0	0	0	1	0	0	0	0	-1	-2	-1	1	2	2	1	0	0	0	0	0
-1	-1	0	1	0	0	0	0	1	0	0	0	0	-1	-2	-1	1	3	2	1	0	0	0	0	0
-0	-0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-3	-1	1	3	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-3	-2	0	3	3	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-2	-1	0	2	3	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	0	1	3	2	0	0	0	0
0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-0	0	0	0	-1	-1	0	1	3	2	0	0	0
0	0	0	0	-1	-1	-1	-1	0	0	0	0	0	1	0	1	1	1	-2	-1	2	3	1	0	0
0	0	0	0	-1	-2	-1	-0	1	1	1	0	1	0	1	0	-2	-3	-1	2	3	1	0	0	0
0	0	0	0	-1	-2	-1	1	1	1	1	0	0	0	0	-1	-3	-2	0	3	3	0	0	0	0
0	0	0	-1	-1	0	1	1	1	0	0	0	0	-1	-3	-2	1	3	2	0	0	0	0	0	0
0	0	0	-1	0	0	0	0	0	0	0	0	0	0	-1	-3	-2	1	3	2	0	0	0	0	0
0	0	0	-1	0	0	0	0	0	0	0	0	0	0	-1	-2	-1	2	3	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	-3	-1	1	3	2	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-3	-1	0	2	2	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-3	-1	2	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	-2	-2	-1	2	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	1	1	1	0	0	0	0	0	0	0

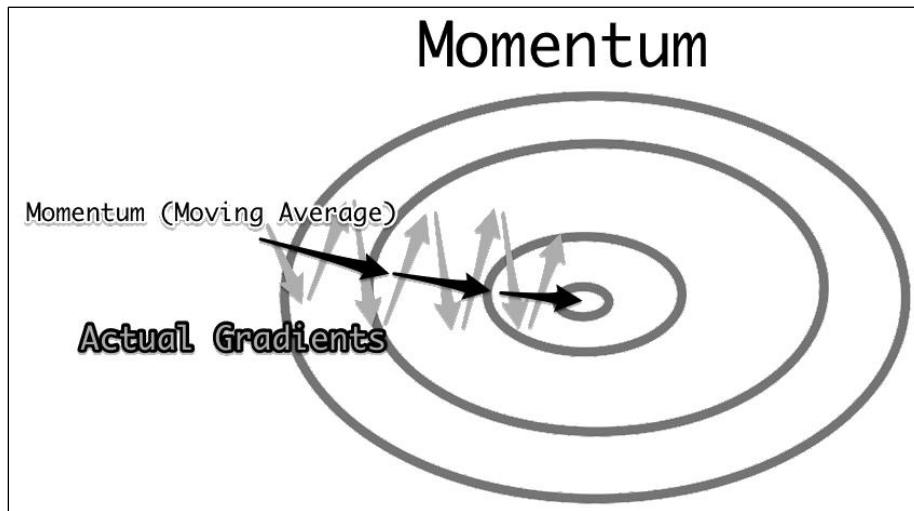
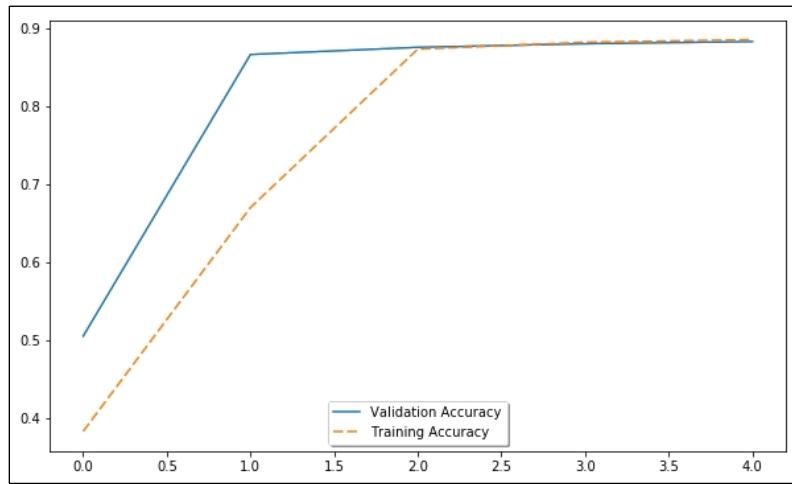
-1	-1	-1
0	0	0
1	1	1

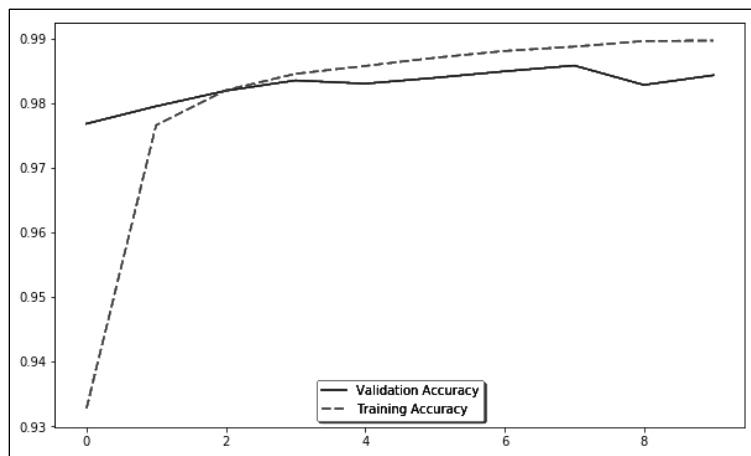
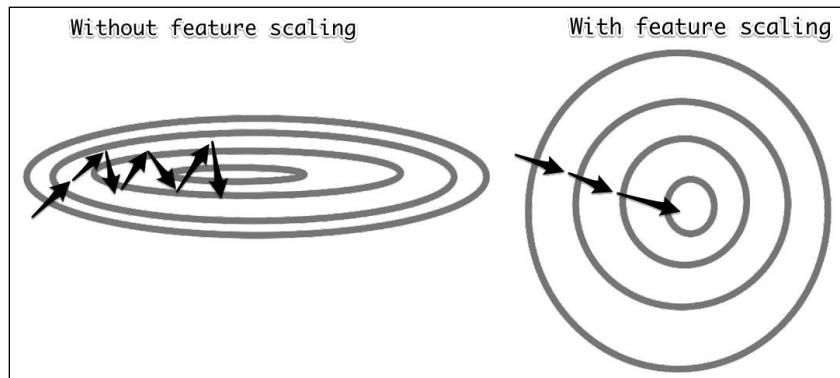
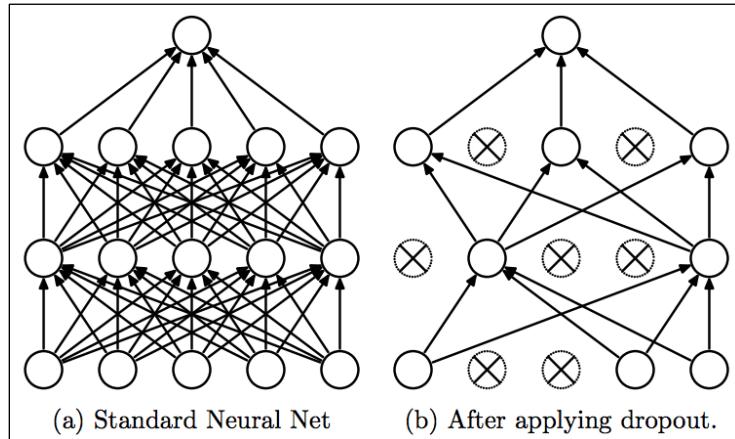


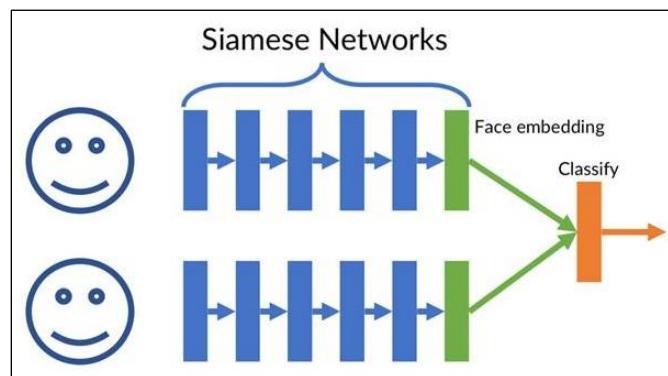
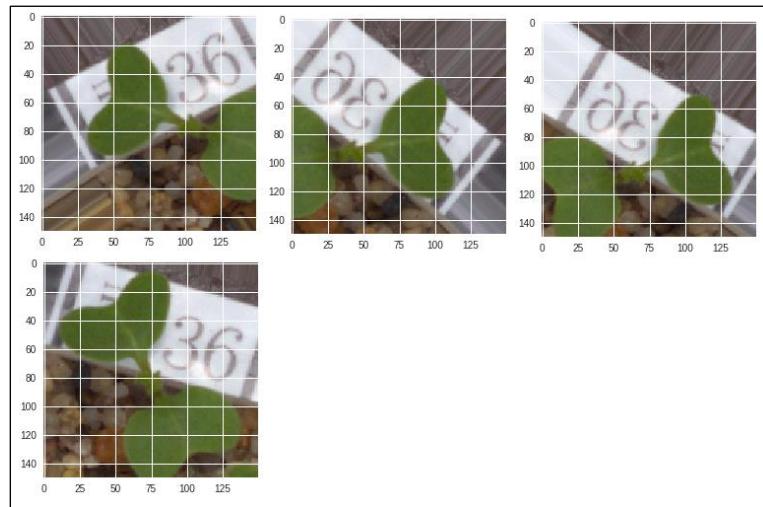




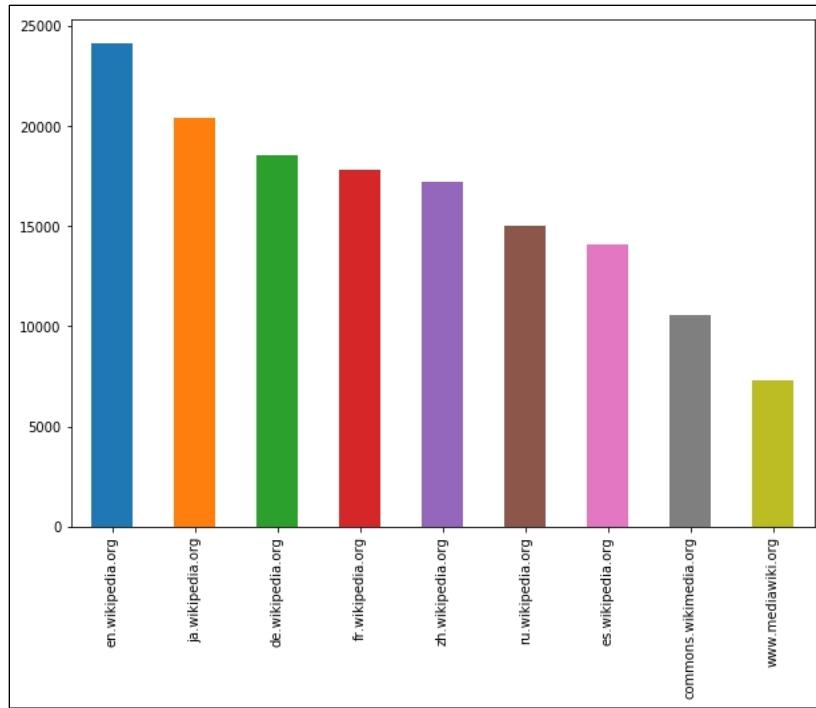
	Input				Output	
	1	2	5	6		
	3	4	2	1	4	6
	2	1	6	5	2	6
	1	1	5	5		
Max Pooling						

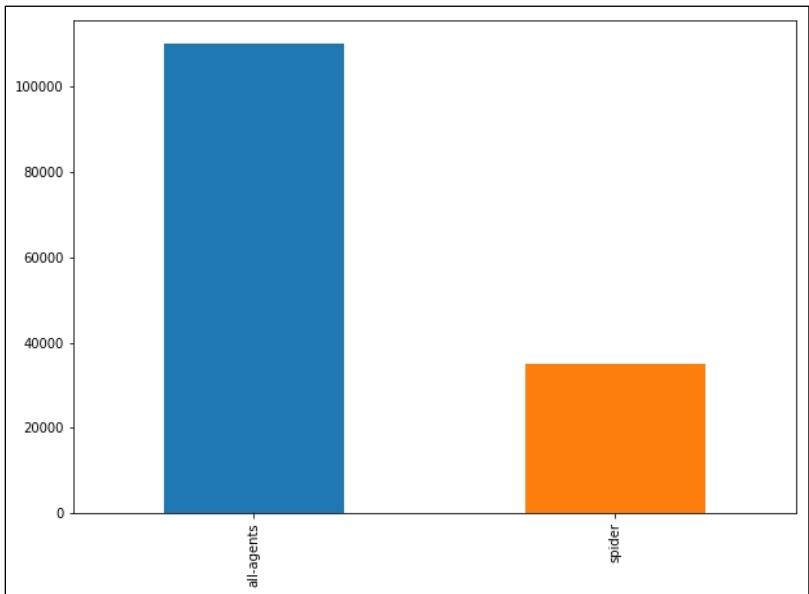
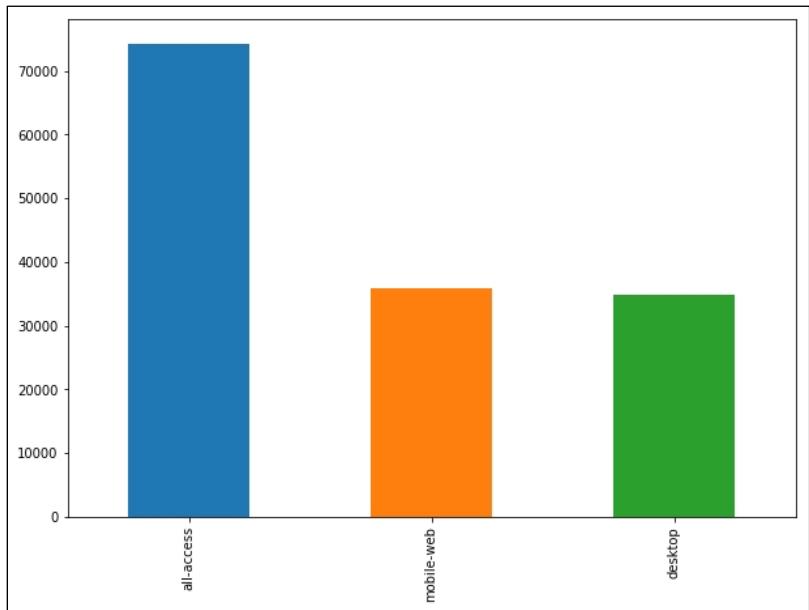


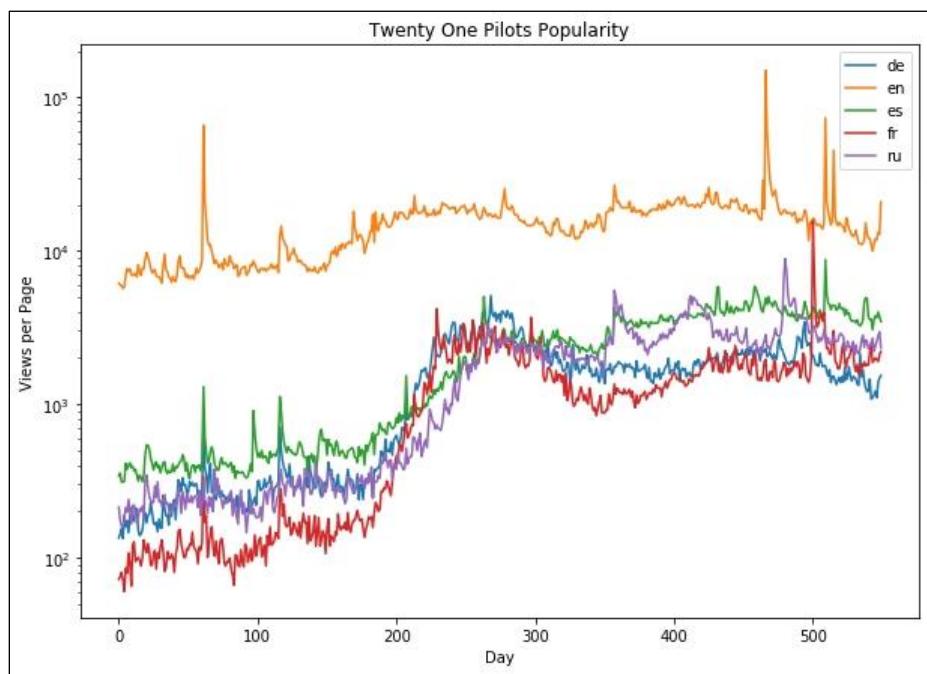
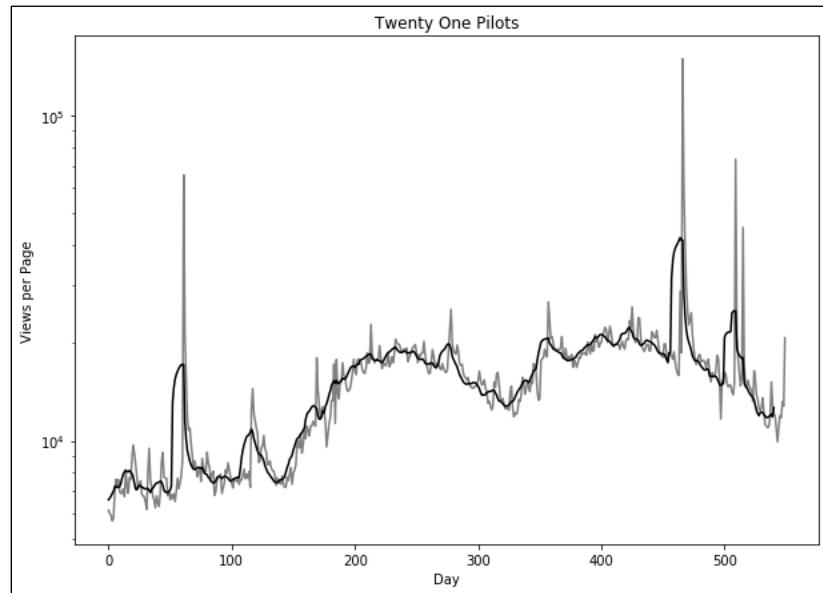


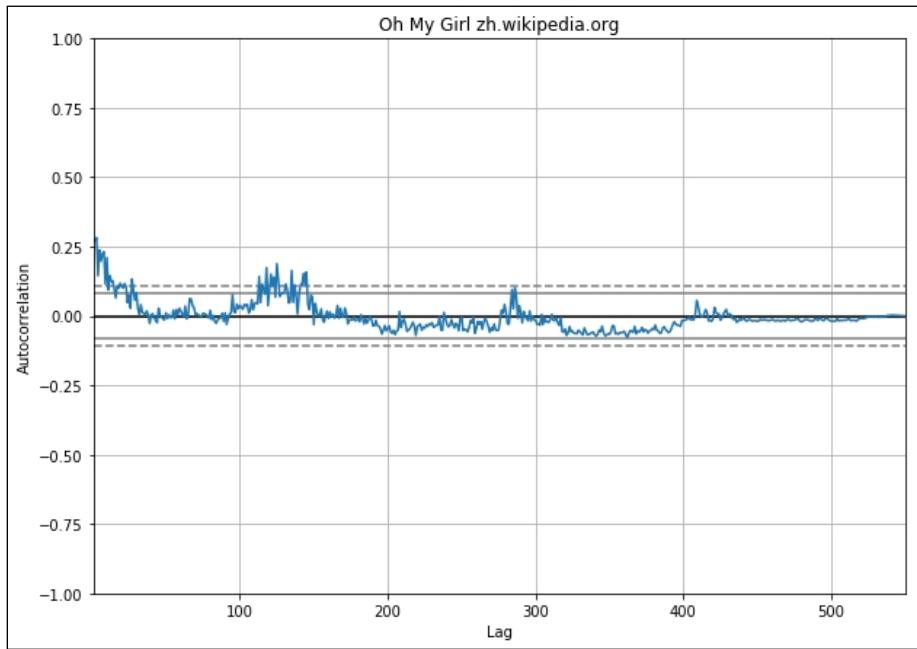
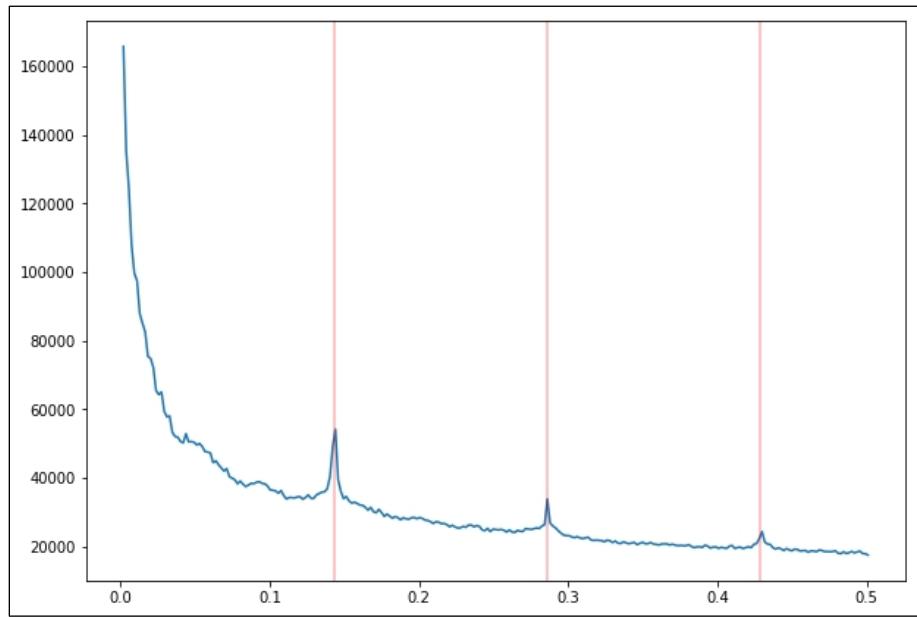


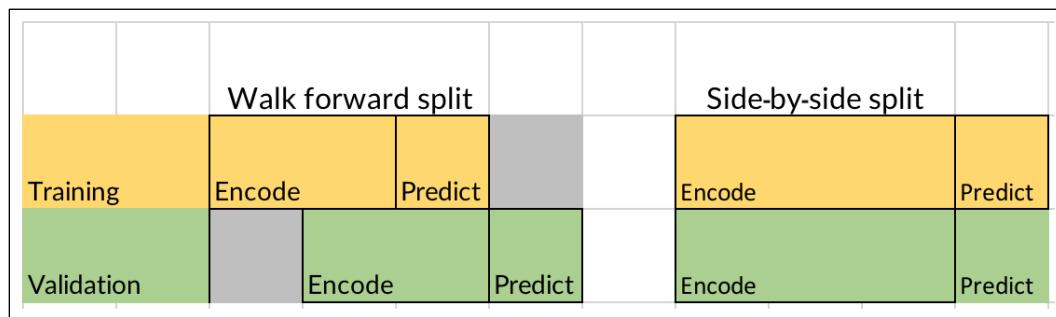
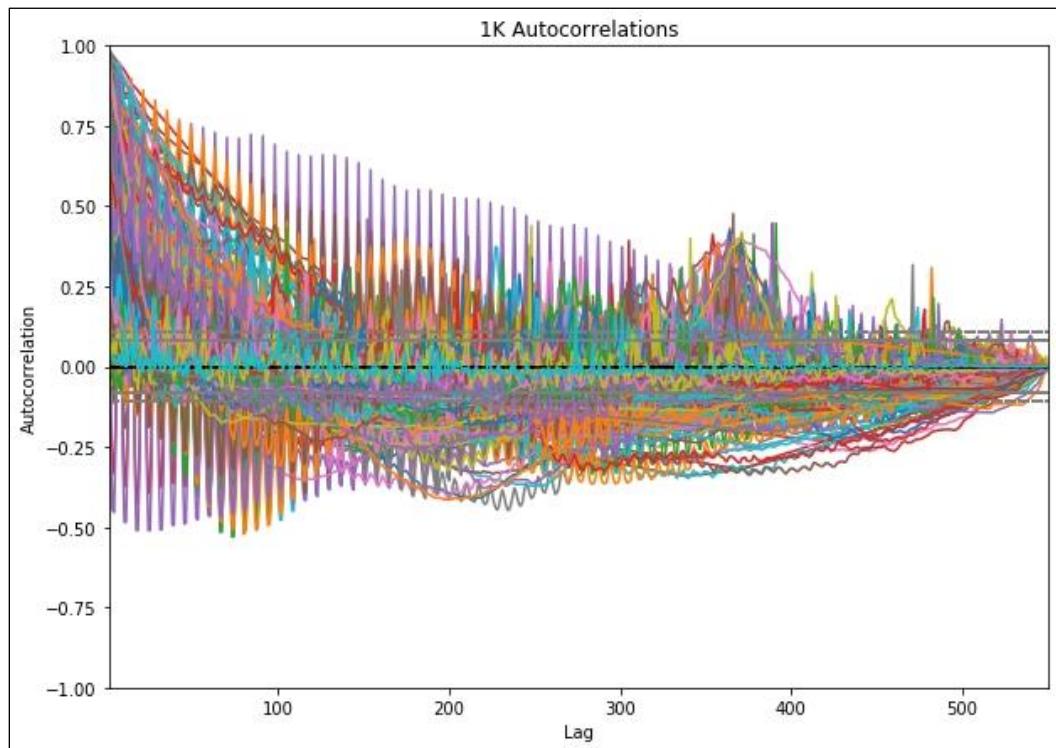
Chapter 4: Understanding Time Series

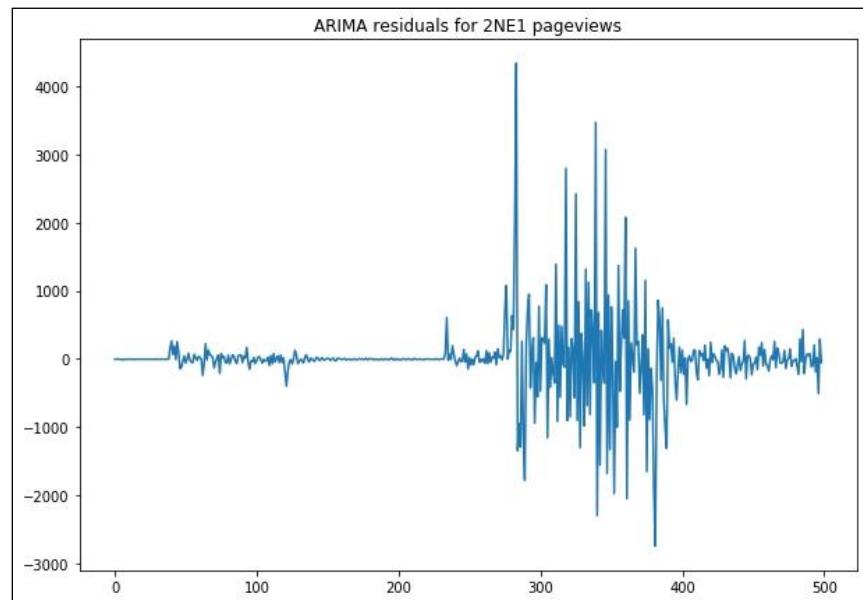
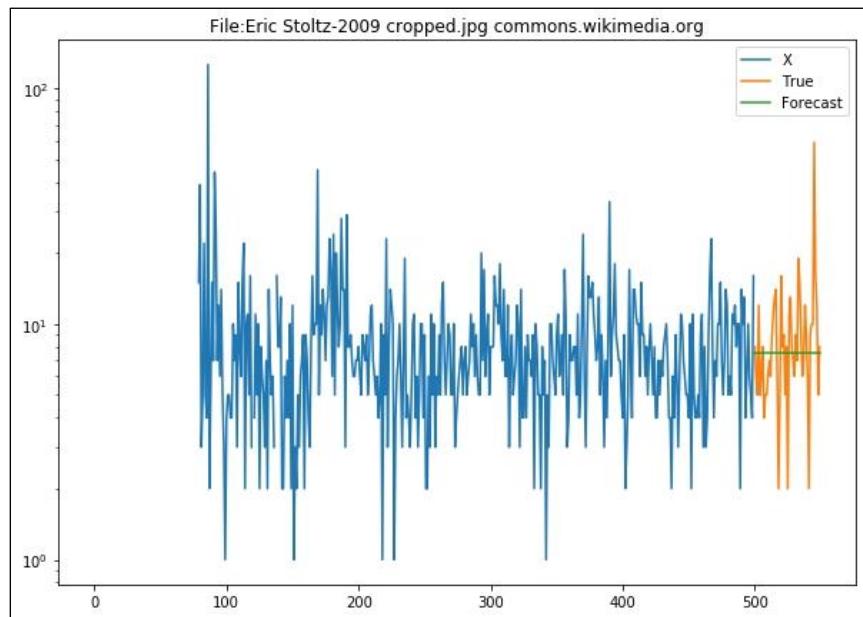


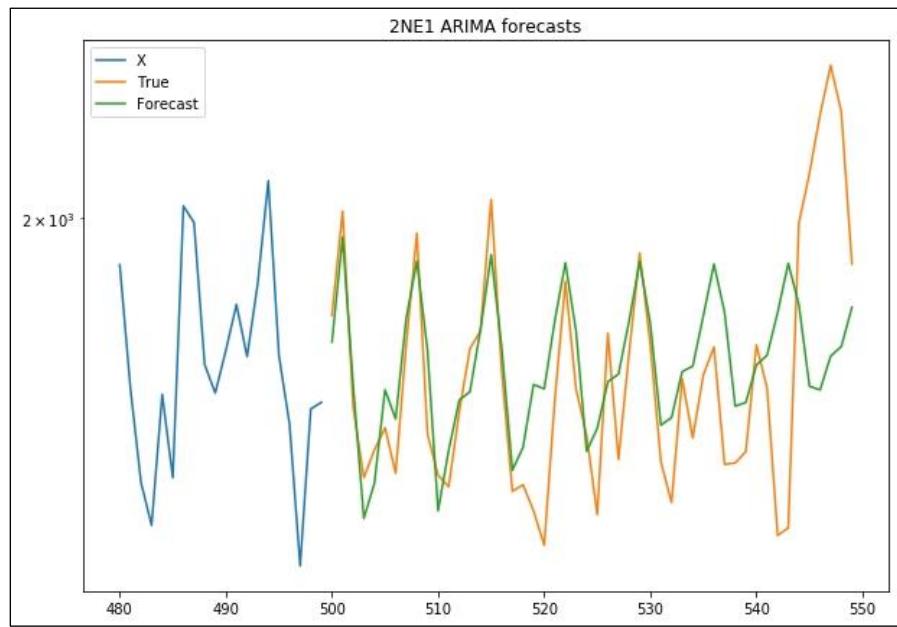
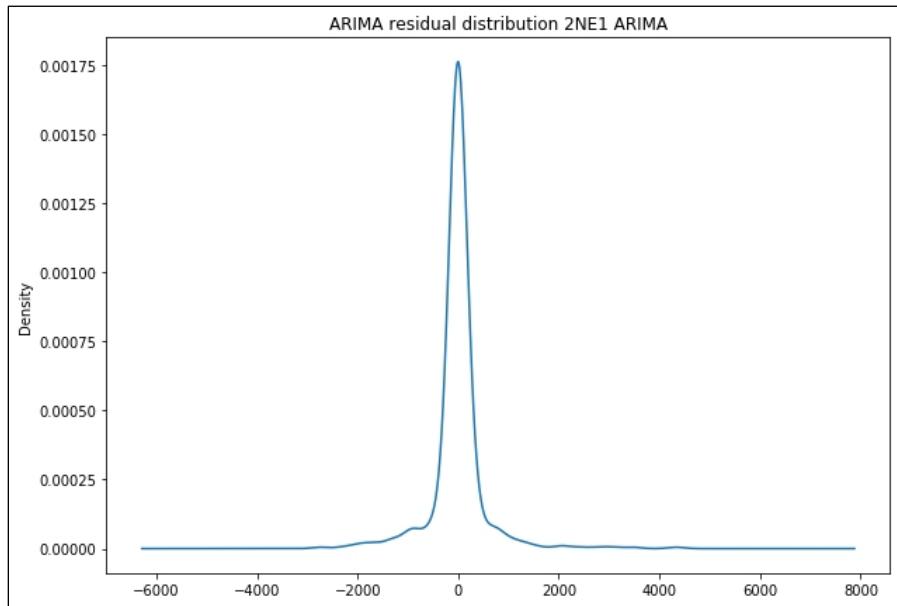


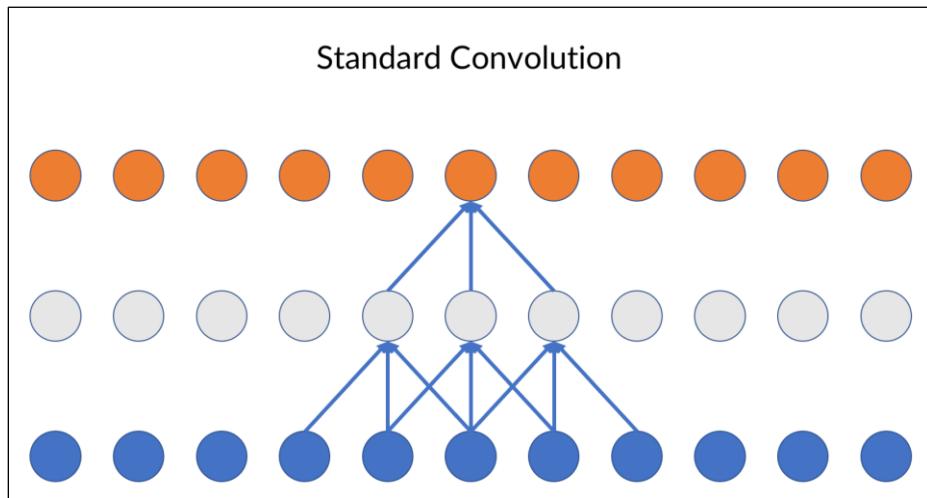
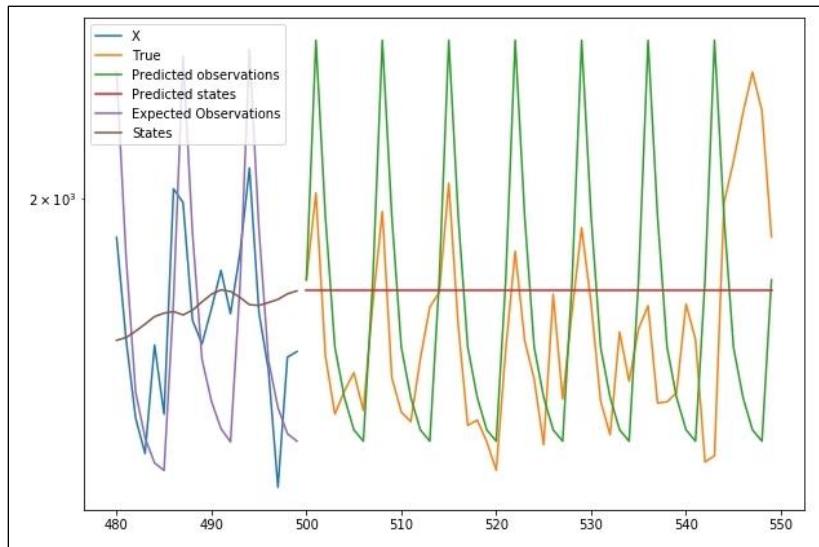


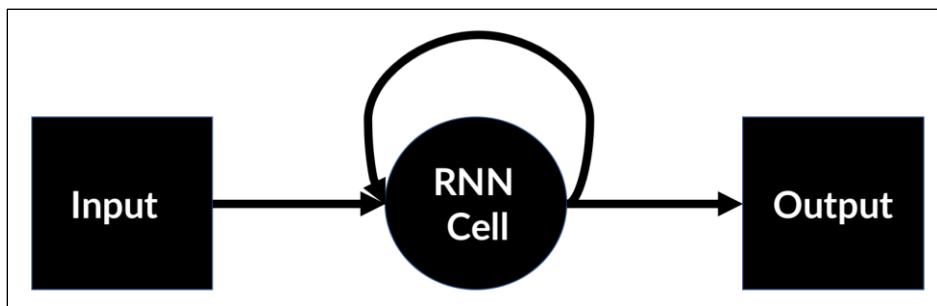
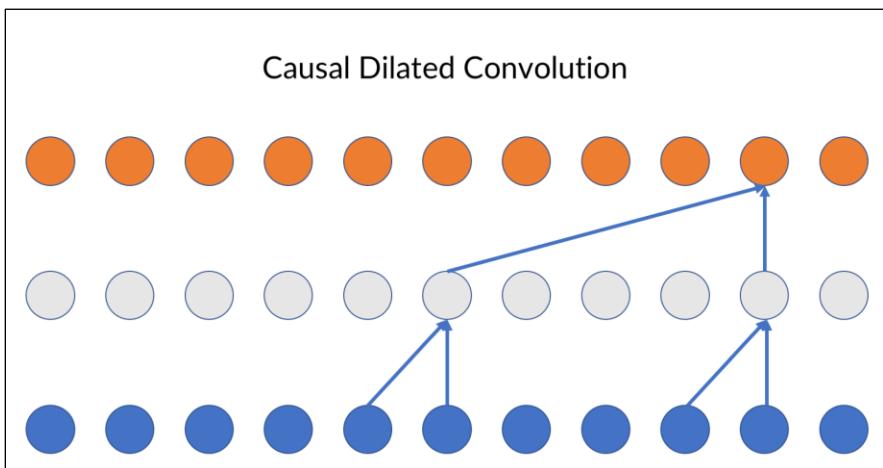
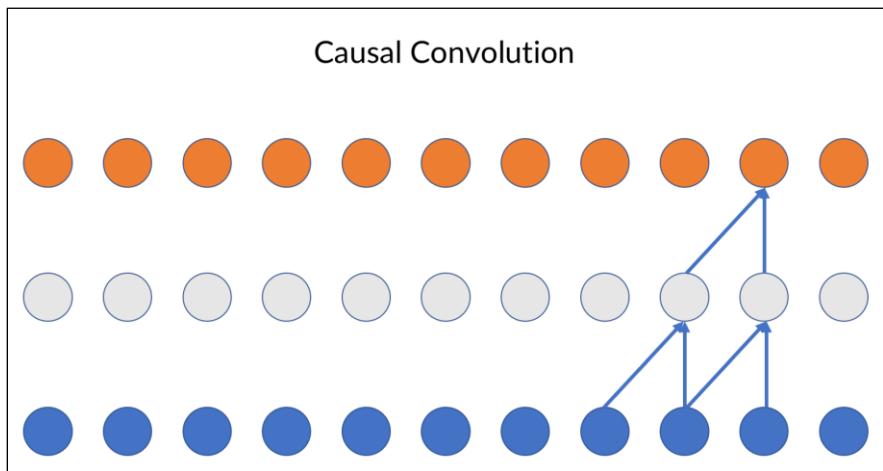


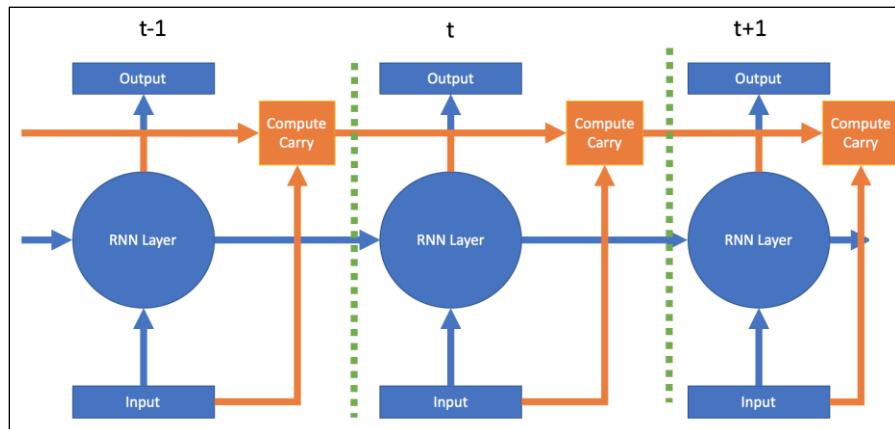
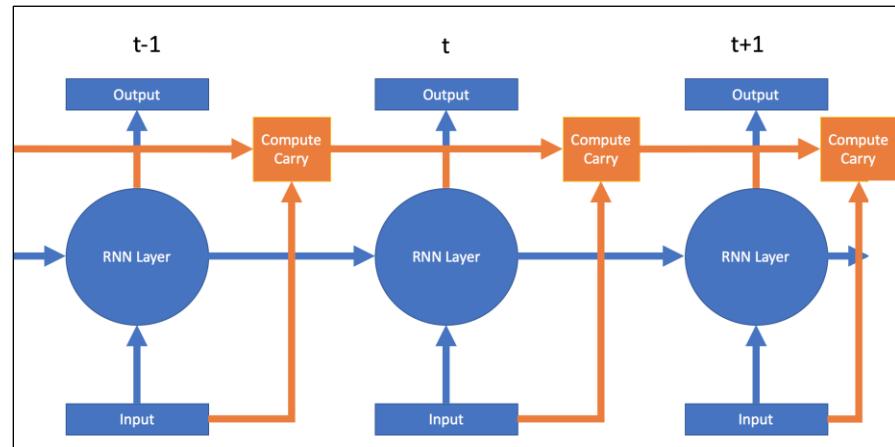
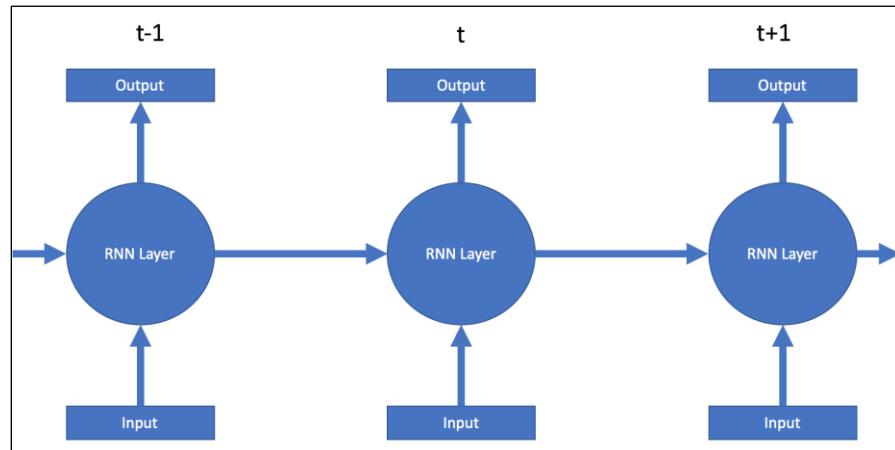


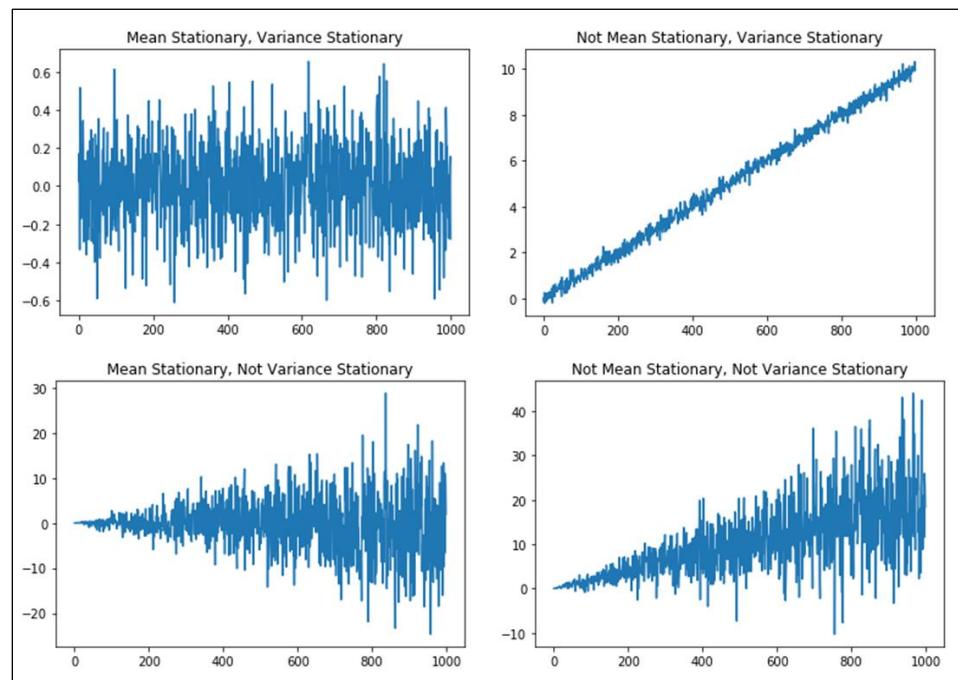
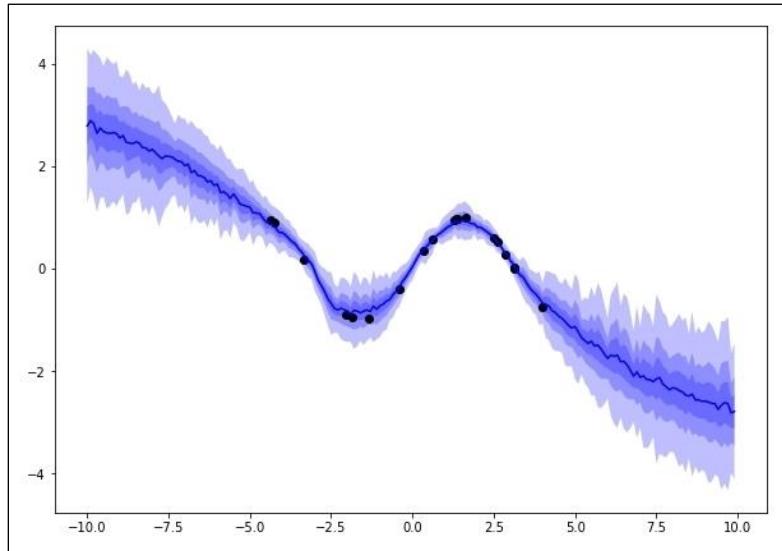


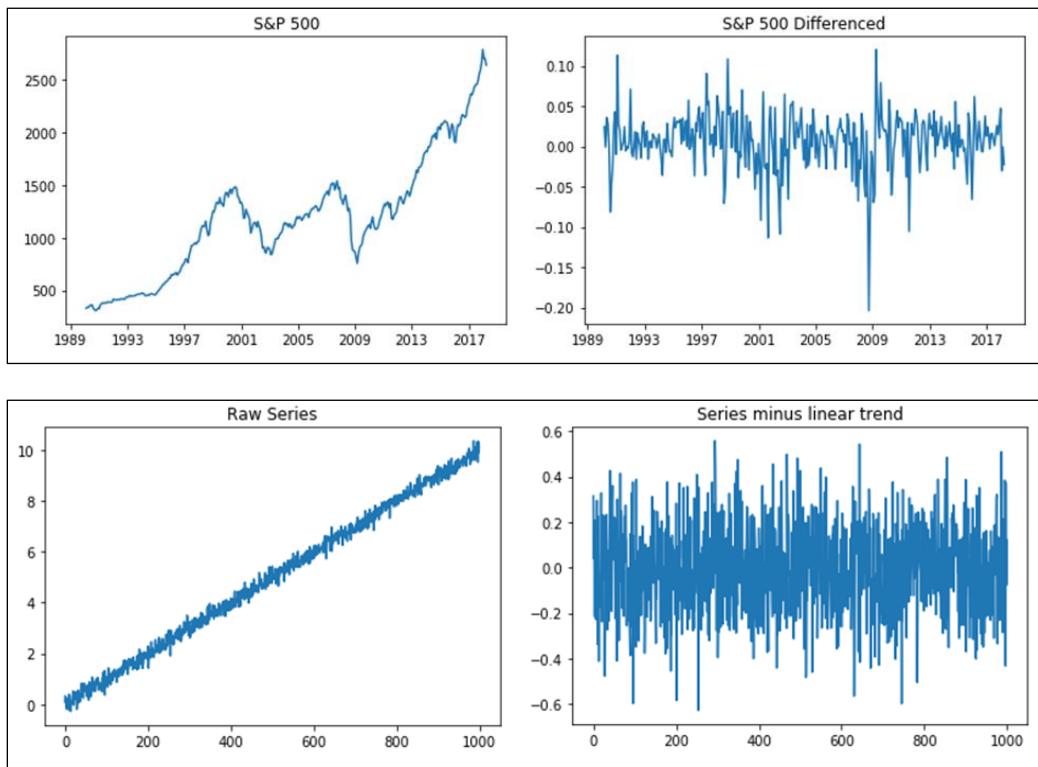




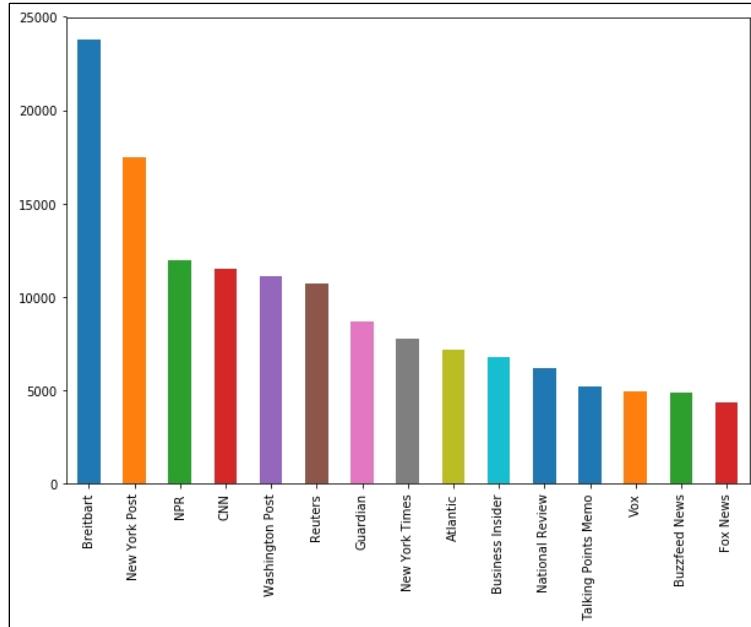




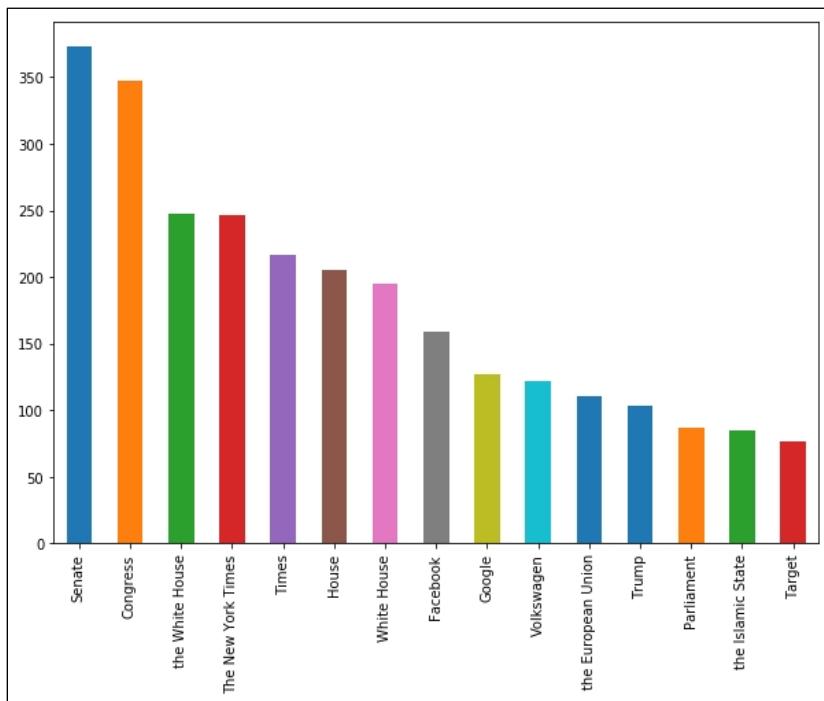
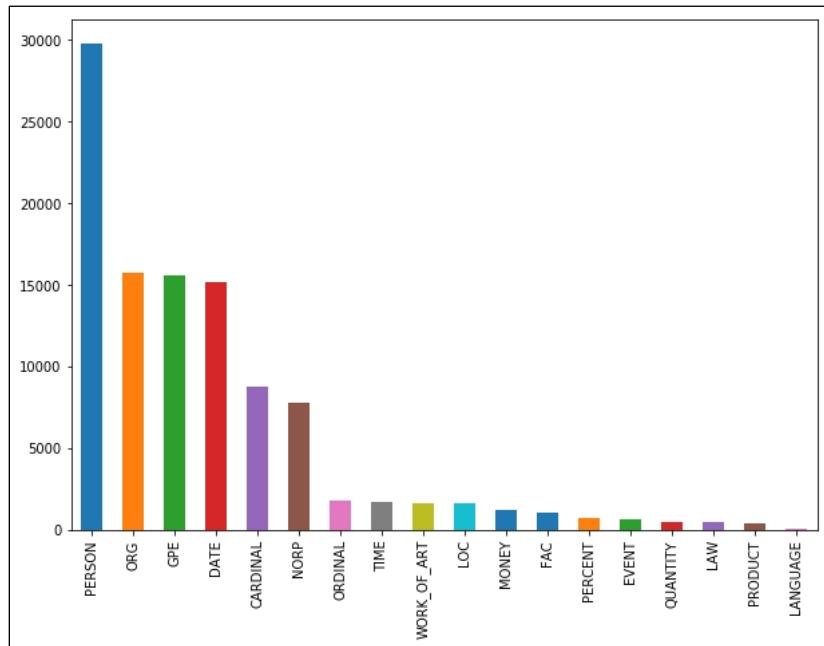


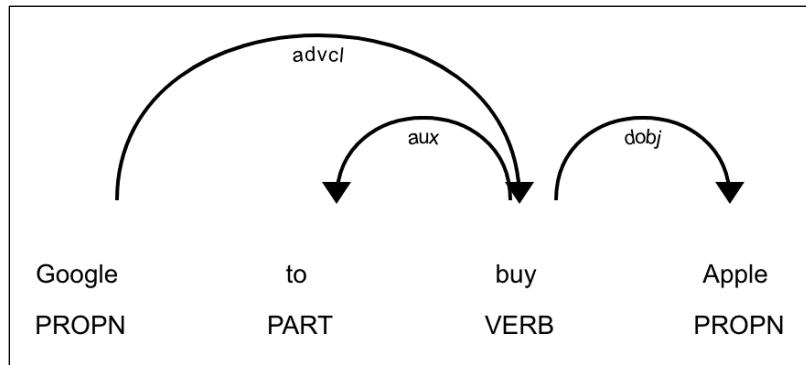


Chapter 5: Parsing Textual Data with Natural Language Processing

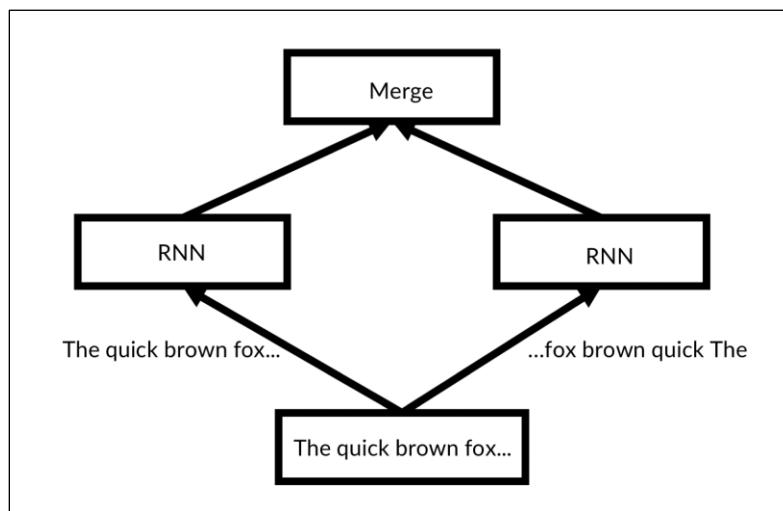


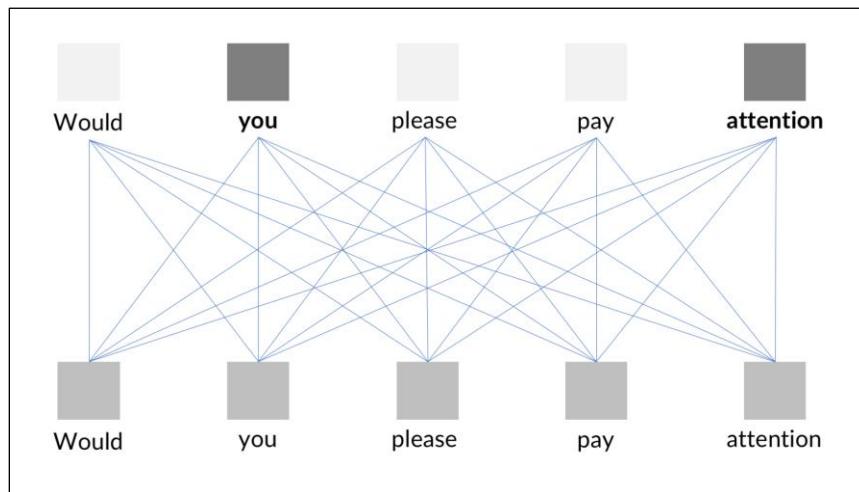
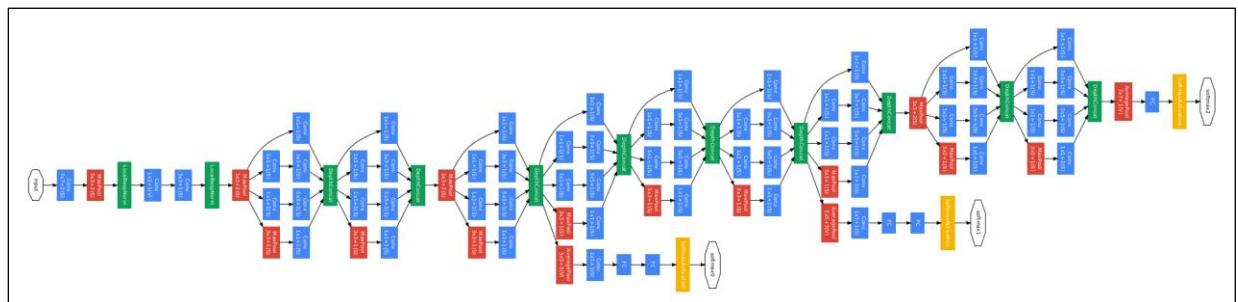
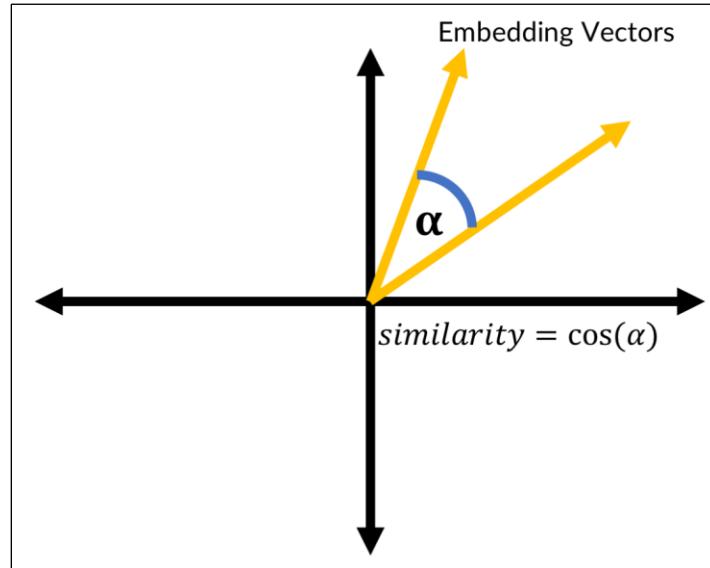
WASHINGTON GPE — Congressional **Republicans NORP** have a new fear when it comes to their health care lawsuit against the **Obama PERSON** administration: They might win. The incoming **Trump NORP** administration could choose to no longer defend the executive branch against the suit, which challenges the administration's authority to spend **billions of dollars MONEY** on health insurance subsidies for **ORG** and **Americans NORP**, handing **House ORG** **Republicans NORP** a big victory on issues. But a sudden loss of the disputed subsidies could conceivably cause the health care program to implode, leaving **millions CARDINAL** of people without access to health insurance before **Republicans NORP** have prepared a replacement. That could lead to chaos in the insurance market and spur a political backlash just as **Republicans NORP** gain full control of the government. To stave off that outcome, **Republicans NORP** could find themselves in the awkward position of appropriating huge sums to temporarily prop up the **Obama GPE** health care law, angering conservative voters who have been demanding an end to the law for **years DATE**. In another twist, **Donald J. Trump's PERSON** administration, worried about preserving executive branch prerogatives, could choose to fight its **Republican NORP** allies in the **House ORG** on some central questions in the dispute. Eager to avoid an ugly political pileup, **Republicans NORP** on **Capitol Hill ORG** and the **Trump NORP** transition team are gazing out how to handle the lawsuit, which, after the election, has been put in limbo until **at least late February DATE** by **the United States Court of Appeals ORG** for **the District of Columbia Circuit GPE**. They are not yet ready to divulge their strategy. "Given that this pending litigation involves the **Obama PERSON** administration and **Congress ORG**, it would be inappropriate to comment," said **Phillip J. Blando PERSON**, a spokesman for the **Trump NORP** transition effort. "Upon taking office, the **Trump NORP** administration will evaluate this case and all related aspects of **the Affordable Care Act LAW**." In a potentially **ORG** decision in **2015 DATE**, Judge **Rosemary M. Collyer PERSON** ruled that **House ORG** **Republicans NORP** had the standing to sue the executive branch over a spending dispute and that the **Obama PERSON** administration had been distributing the health insurance subsidies, in violation of the **Constitution LAW**, without approval from **Congress ORG**. The **Justice Department ORG**, confident that Judge **Collyer PERSON**'s decision would be reversed, quickly appealed, and the subsidies have remained in place during the appeal. In successfully seeking a temporary halt in the proceedings after Mr. **Trump PERSON** won, **House ORG** **Republicans NORP** last month **DATE** told the court that they "and the 's transition team currently are discussing potential options for resolution of this matter, to take effect after the **NORP**'s inauguration on **Jan. 20, 2017 DATE**." The suspension of the case, **House ORG** lawyers said, will "provide the **ORG** and his future administration time to consider whether to continue prosecuting or to otherwise resolve this appeal." **Republican NORP** leadership officials in the **House ORG** acknowledge the possibility of "cascading effects" if the **ORG** payments, which have totaled **an estimated \$13 billion MONEY**, are

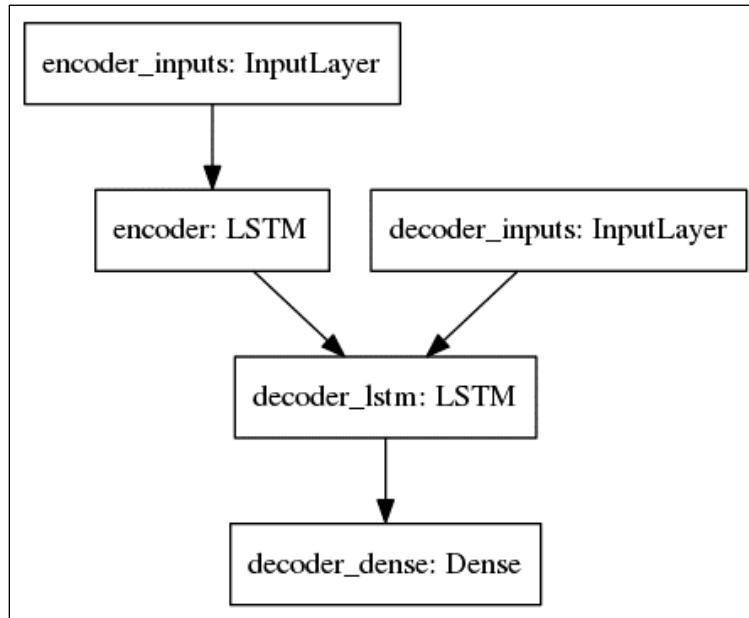
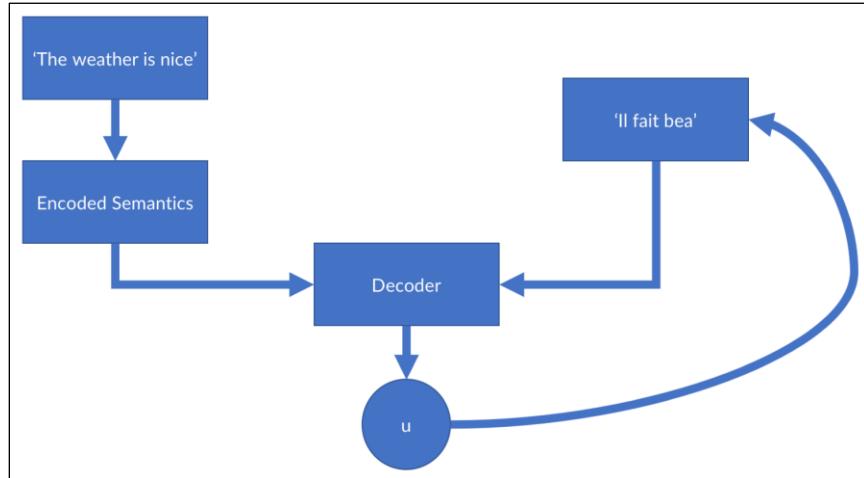
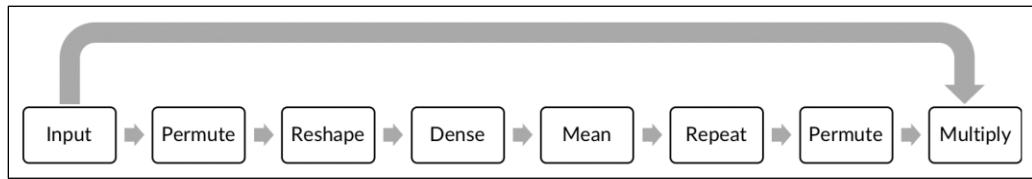


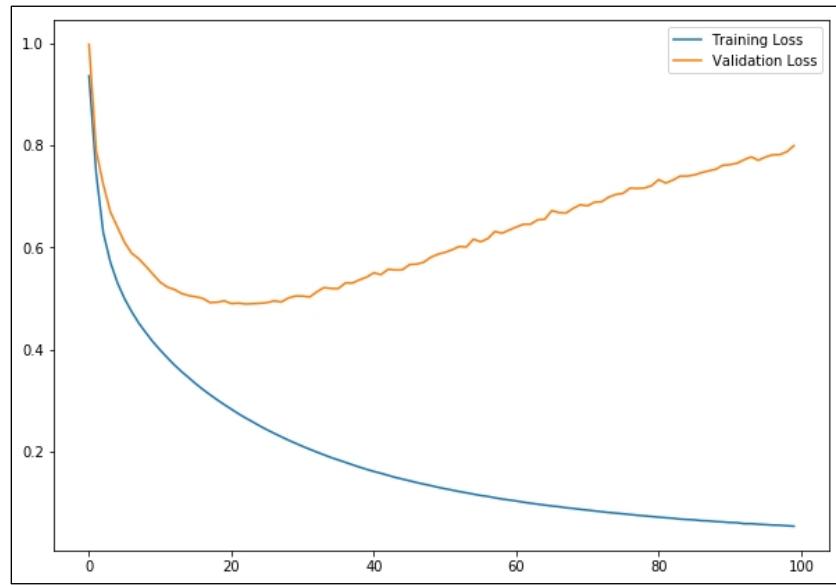


services, as well as uncommon surfaces on which to enlarge photos for display, be it burlap, wood boards, acrylic or fabric. Why not try some fresh sites and methods? I recently sent some **ORG** quality **iPhone PRODUCT** vacation photos to a handful of companies that I'd never used before and had them enlarged to various sizes and printed on different surfaces. I've also offered some guidance about bulk digitizing those boxes of old travel photos sitting in your closet or basement so that you can begin **the New Year EVENT** if not with a vacation, then with a **ORG** home. Of all the ways to turn photos into wall art, I was most interested in trying engineer prints, named for the large, lightweight prints used by architects. For less than the cost of a couple of movie tickets, you can make huge enlargements. Mind you, it's a particular aesthetic, one that's most likely to appeal to people who are after an industrial, shabby chic or bohemian look. The paper is thin and the

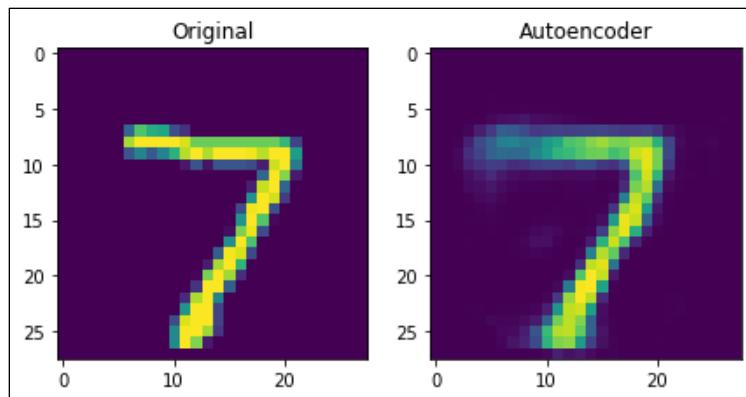
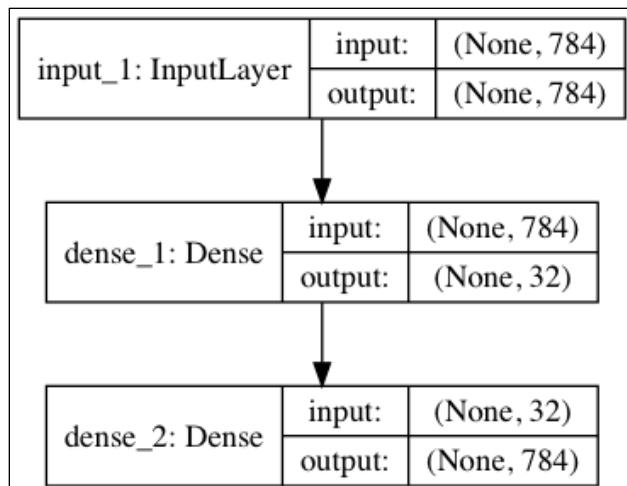
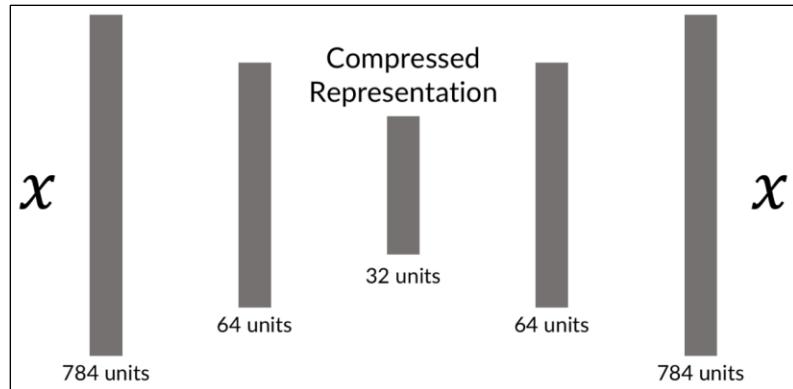


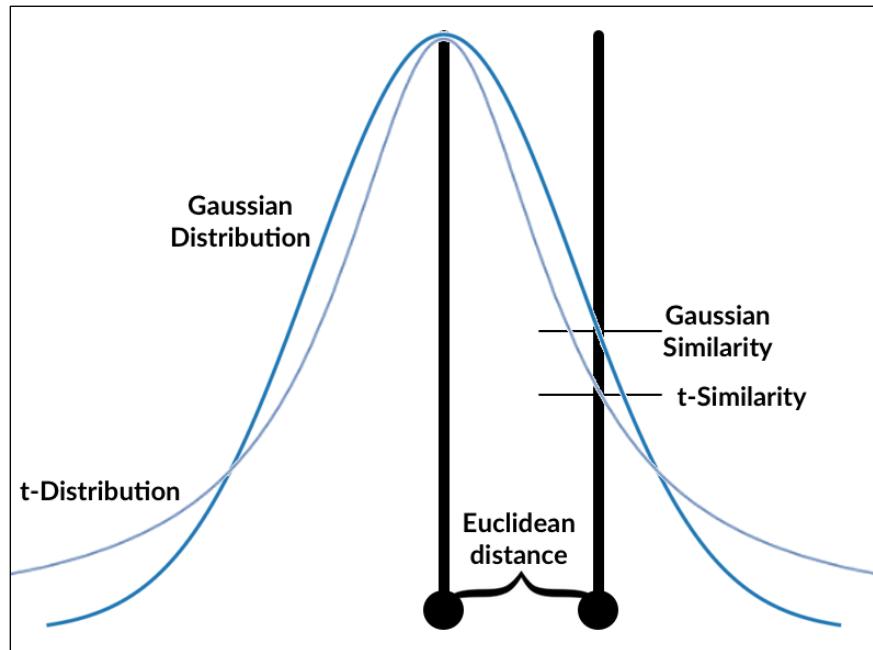
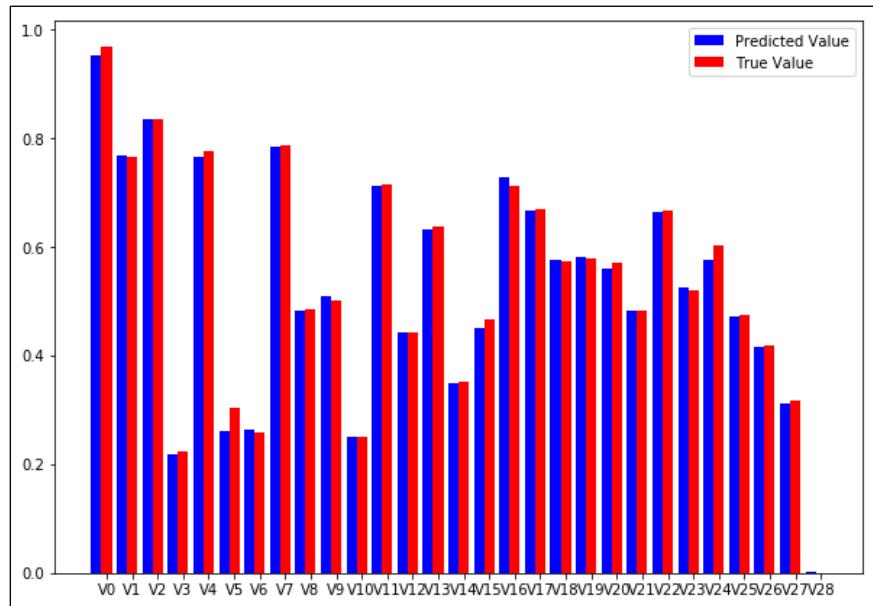


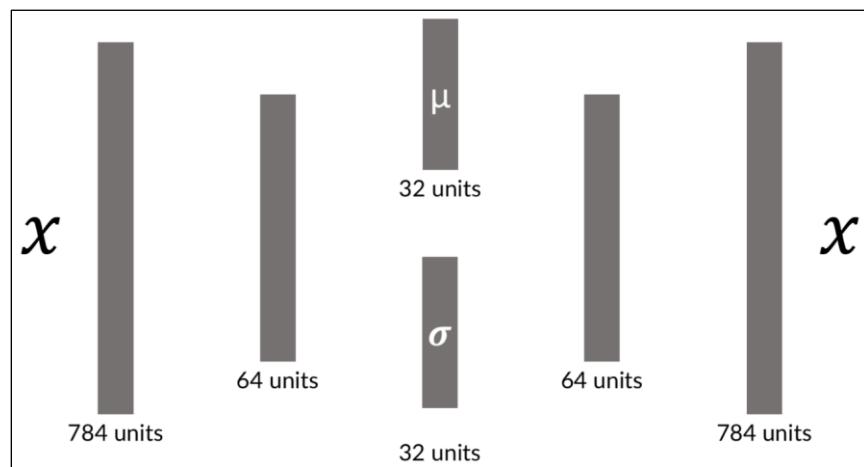
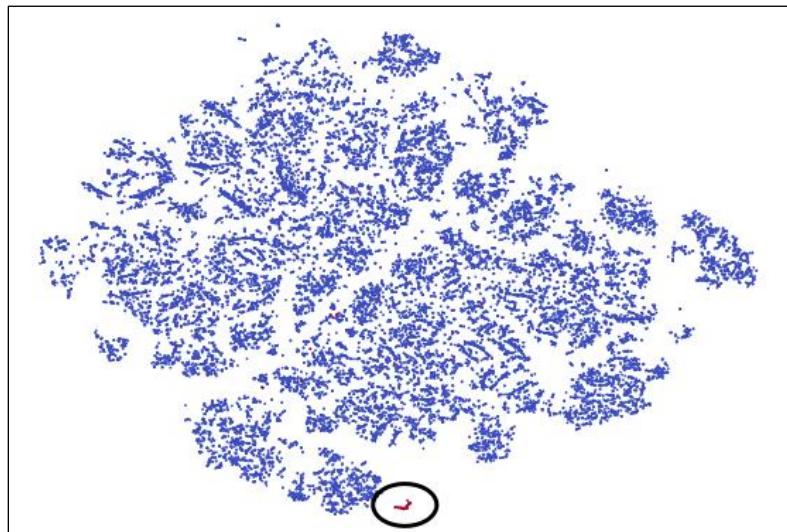


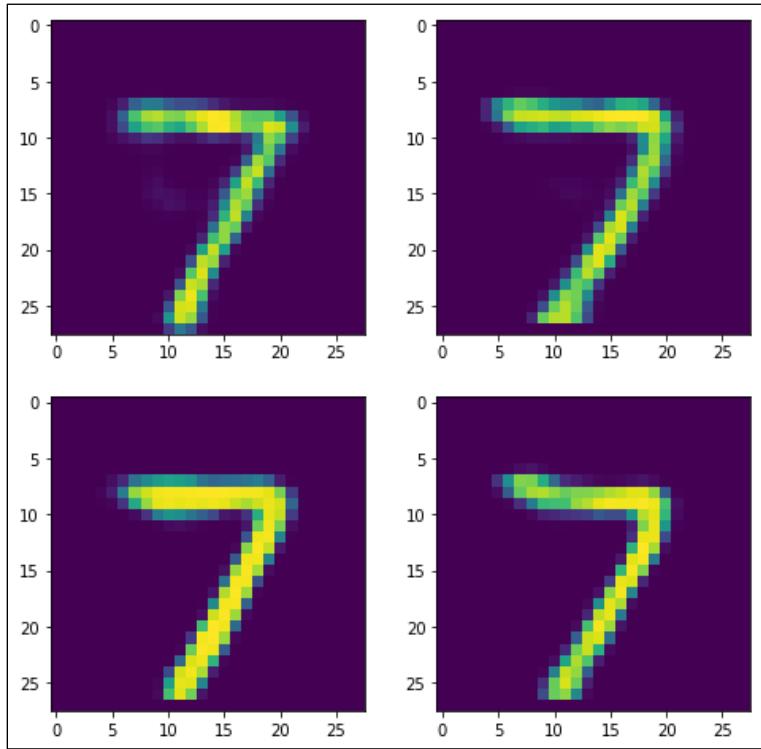
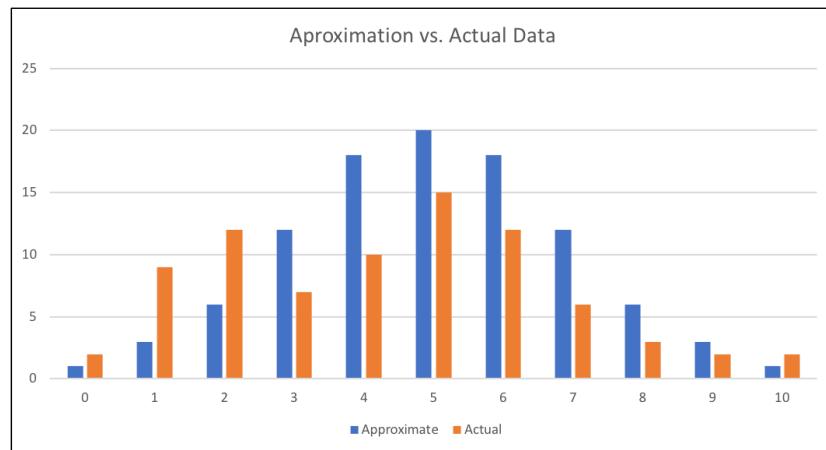


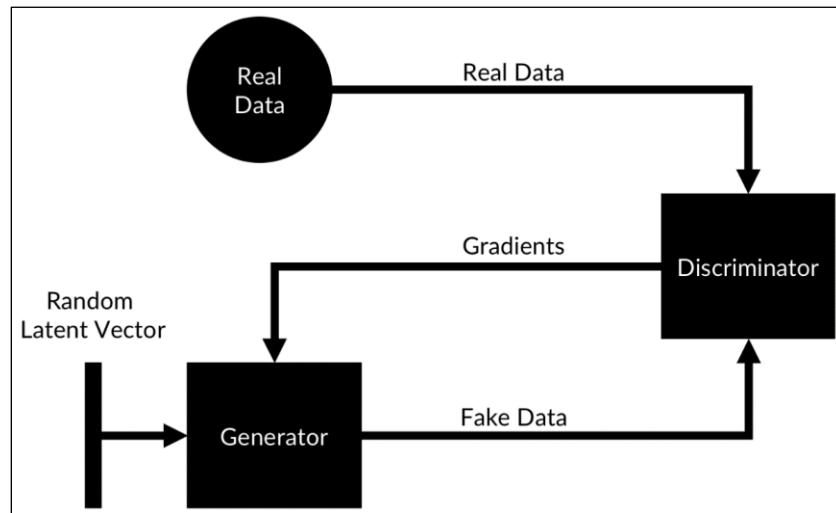
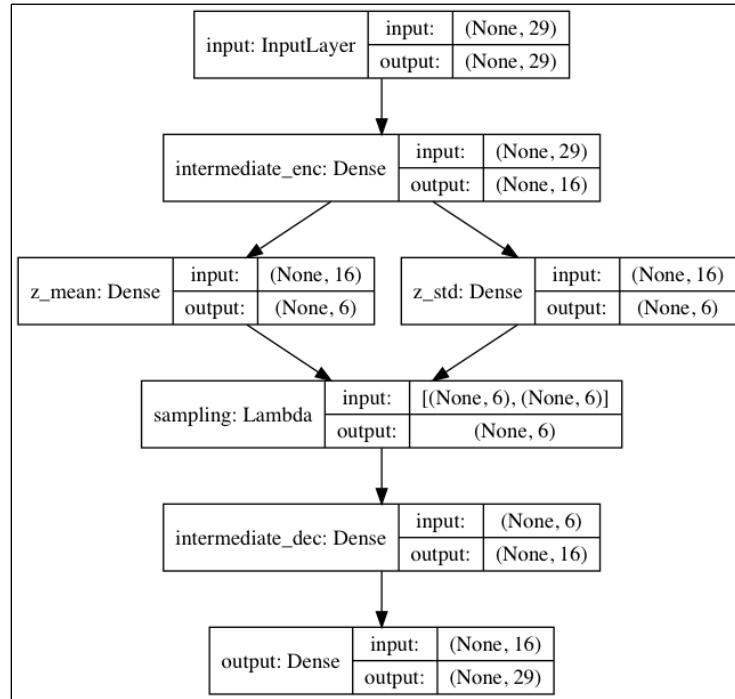
Chapter 6: Using Generative Models

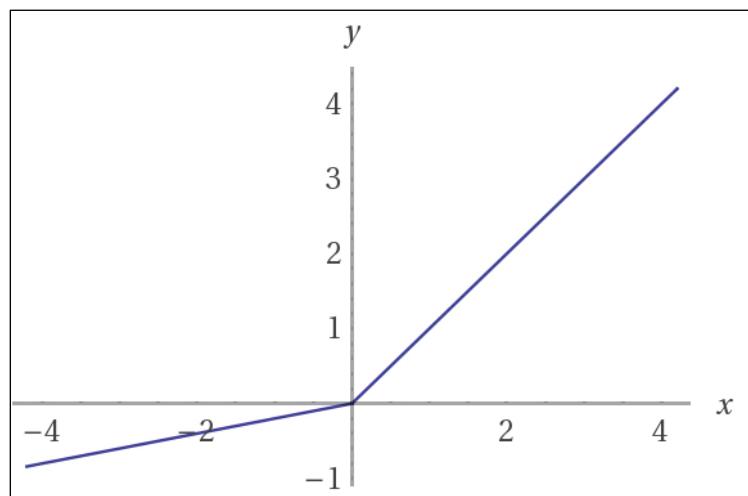




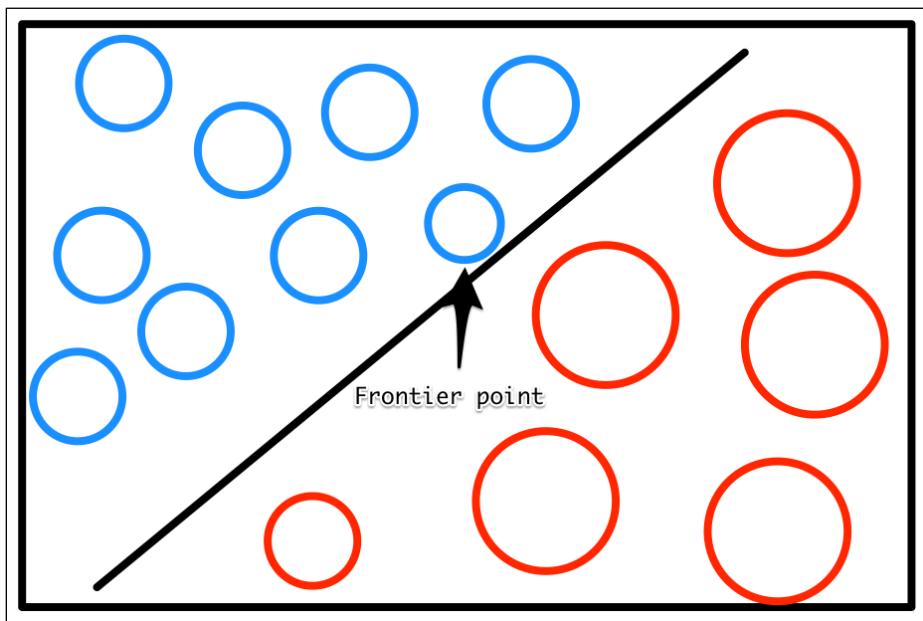
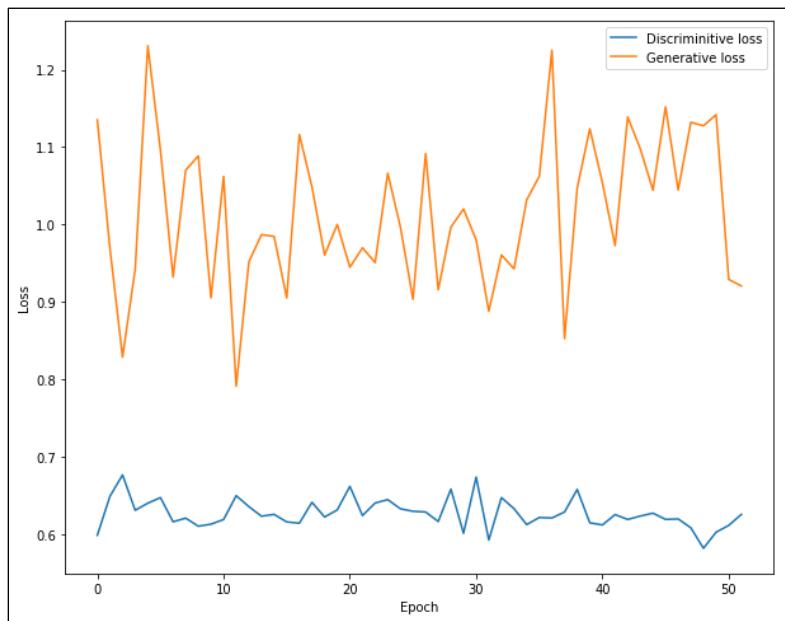


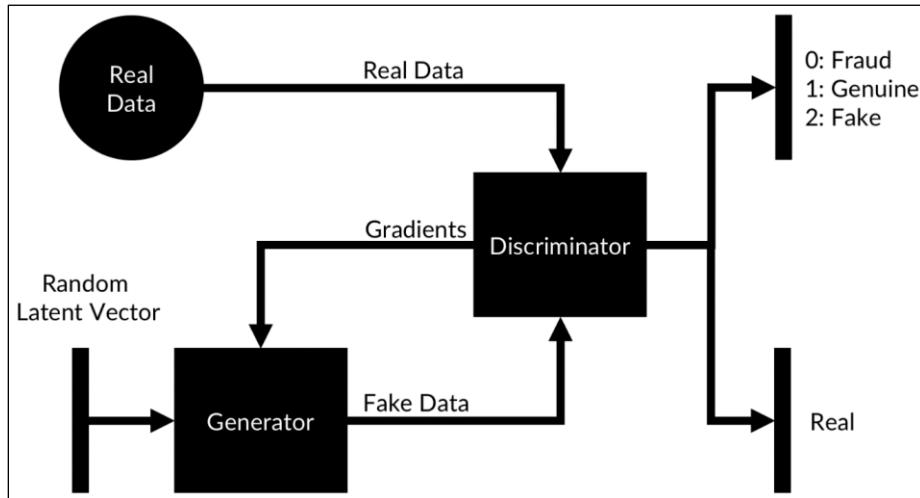
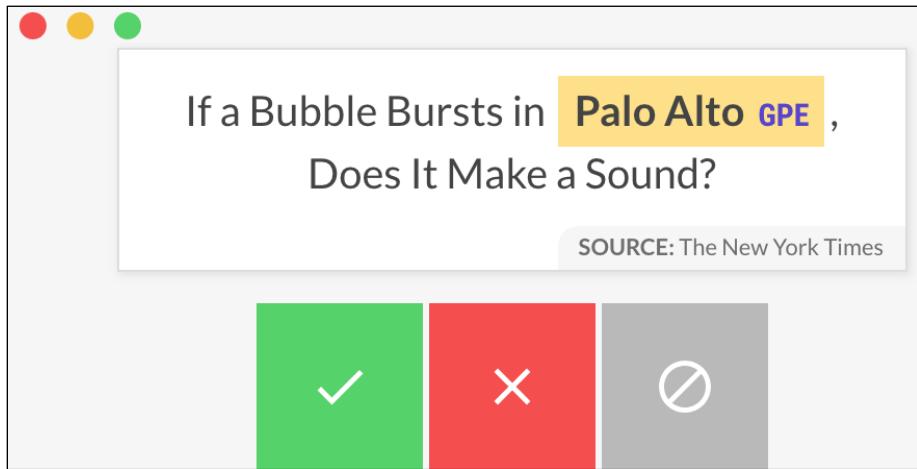


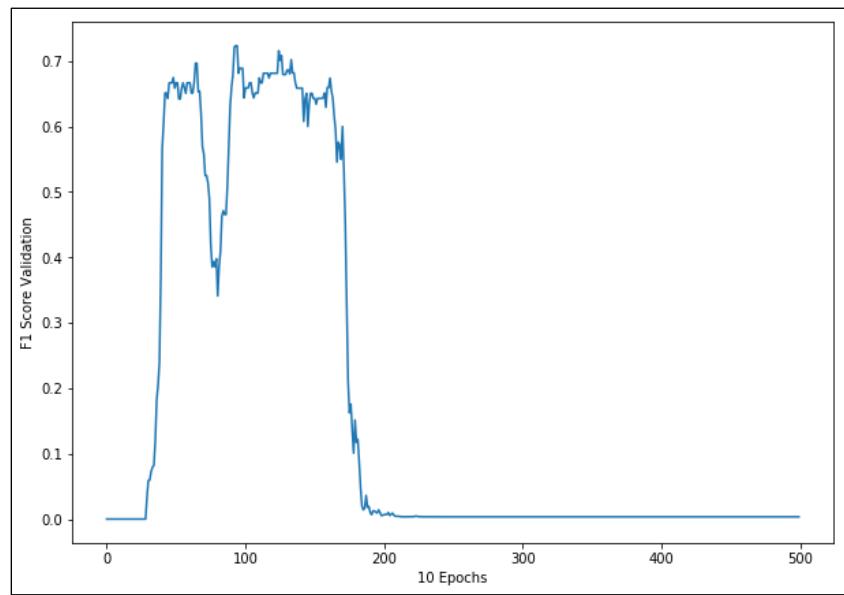




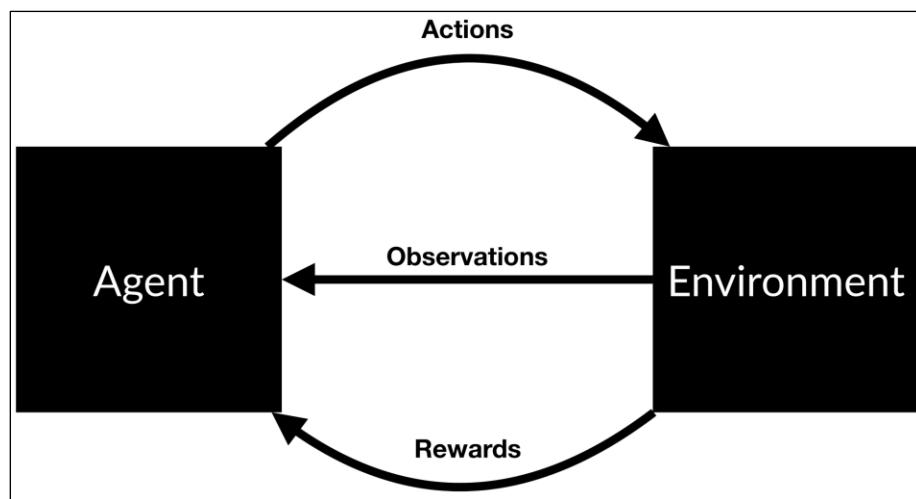
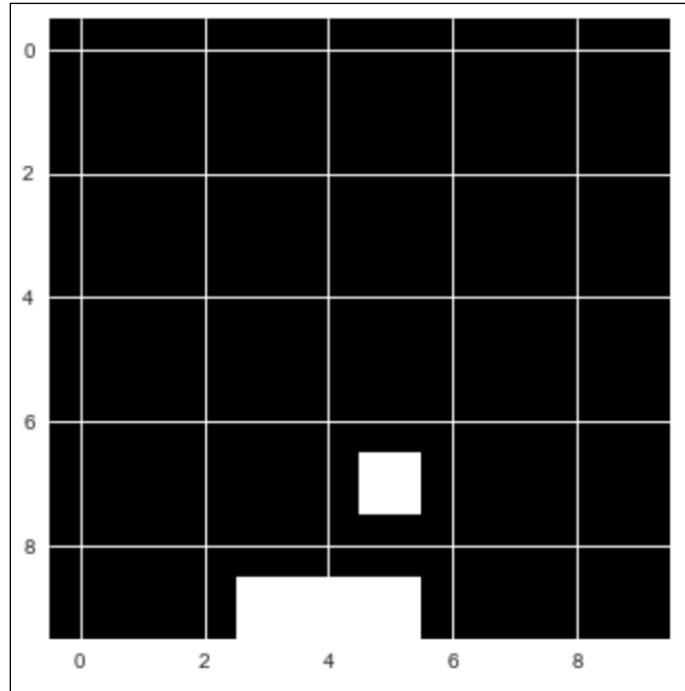
7	4	4	9
7	0	3	7
7	1	6	6
1	4	9	5

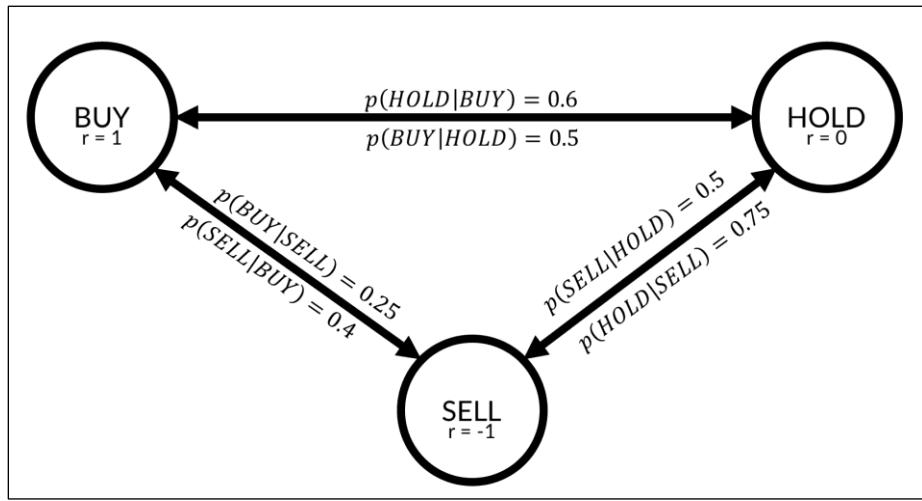
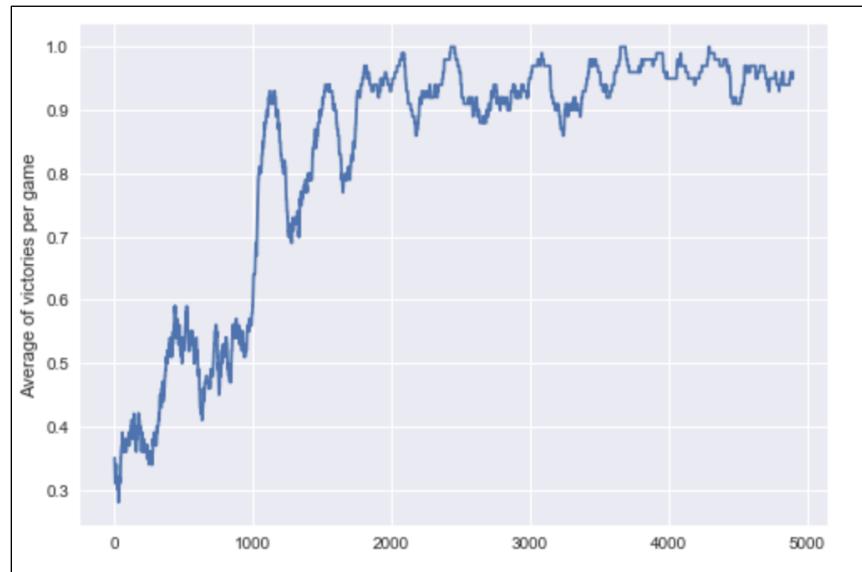


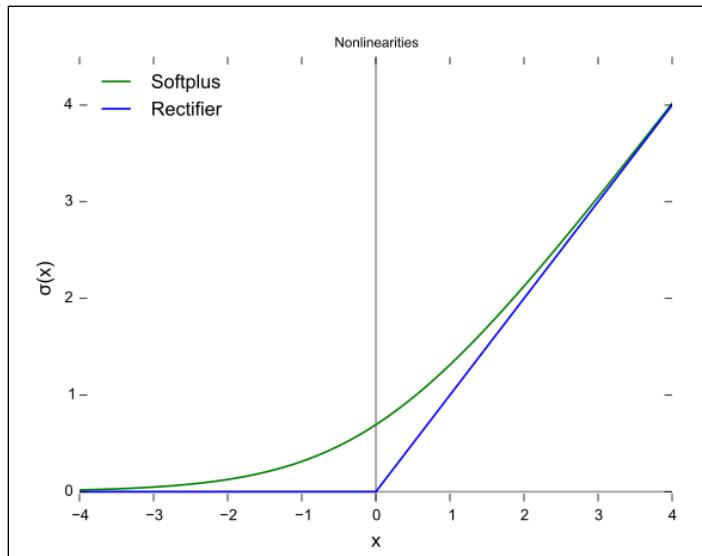
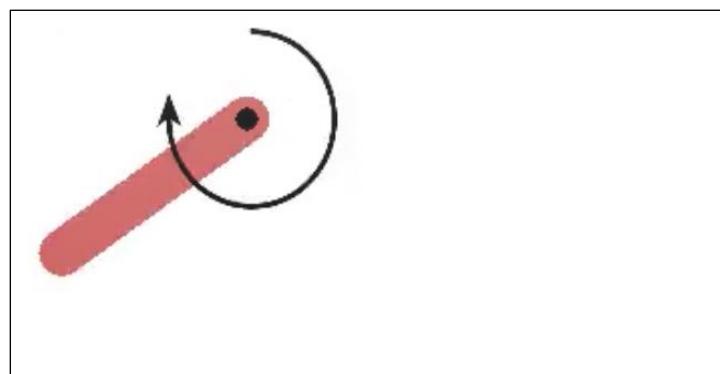
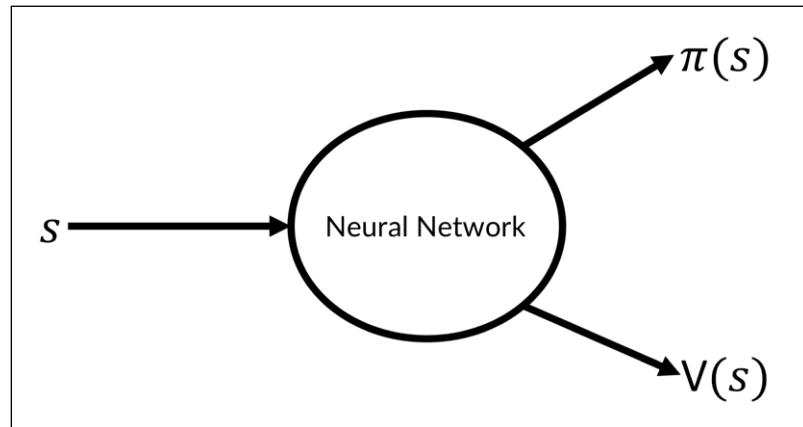


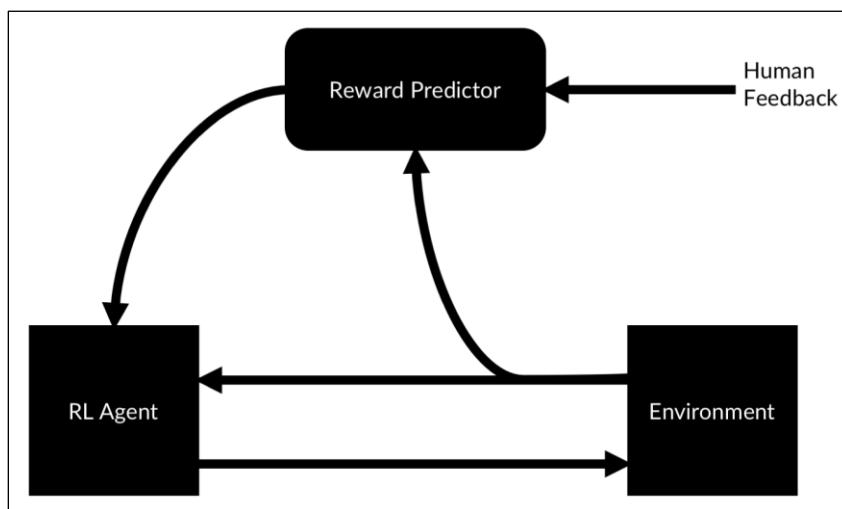
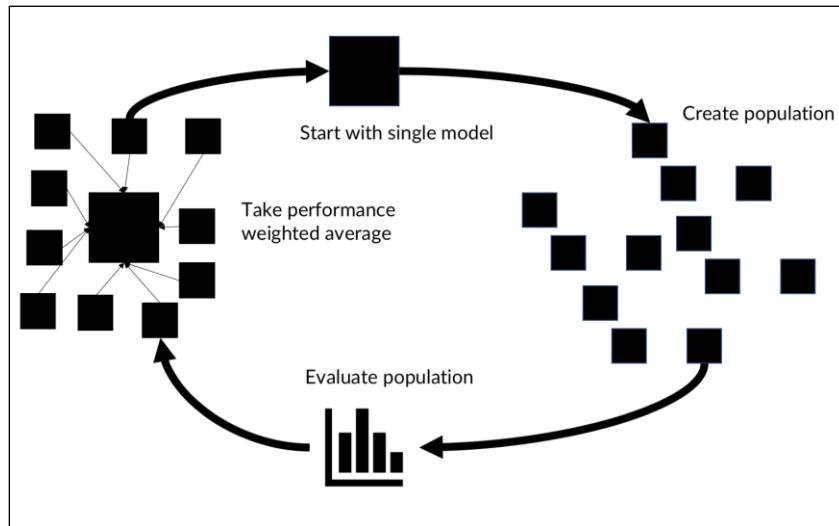


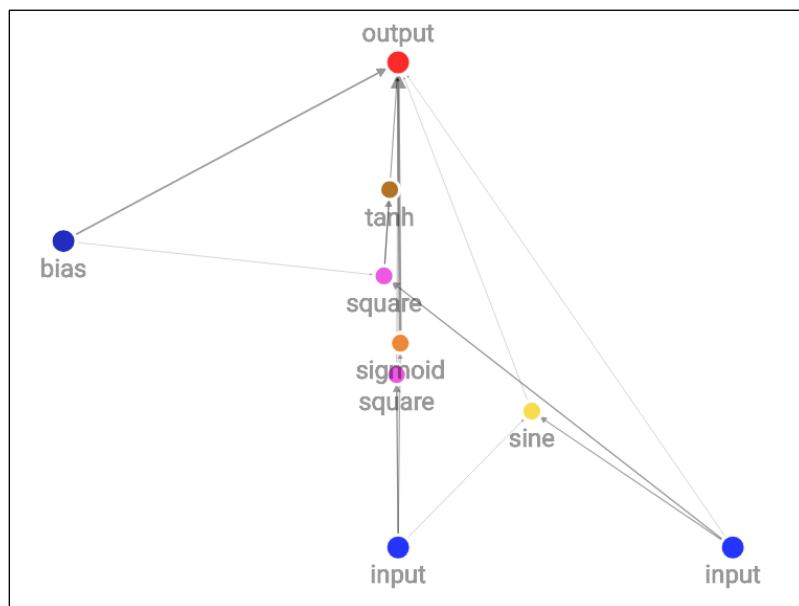
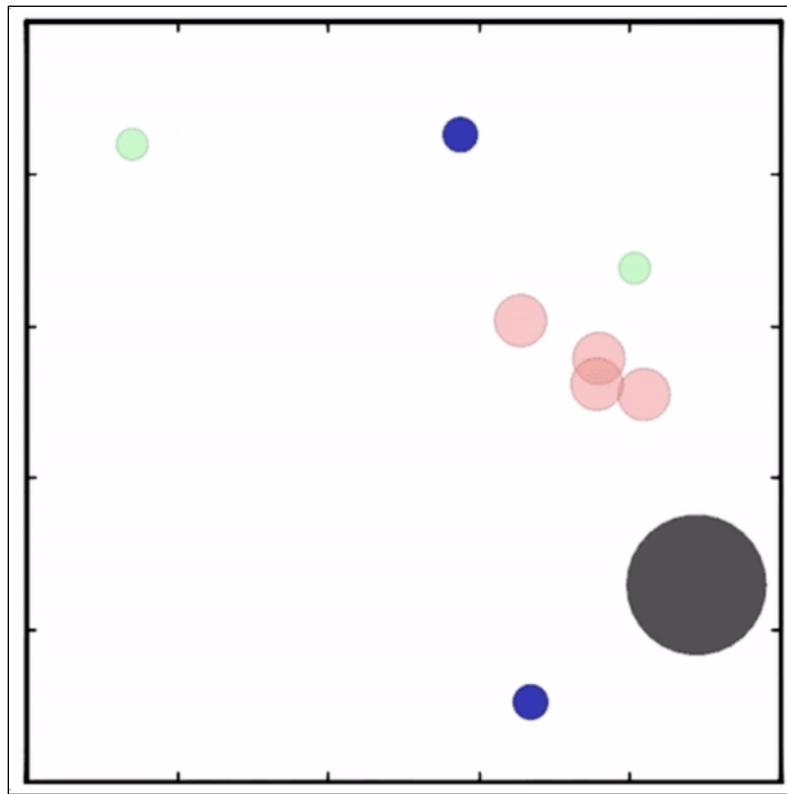
Chapter 7: Reinforcement Learning for Financial Markets



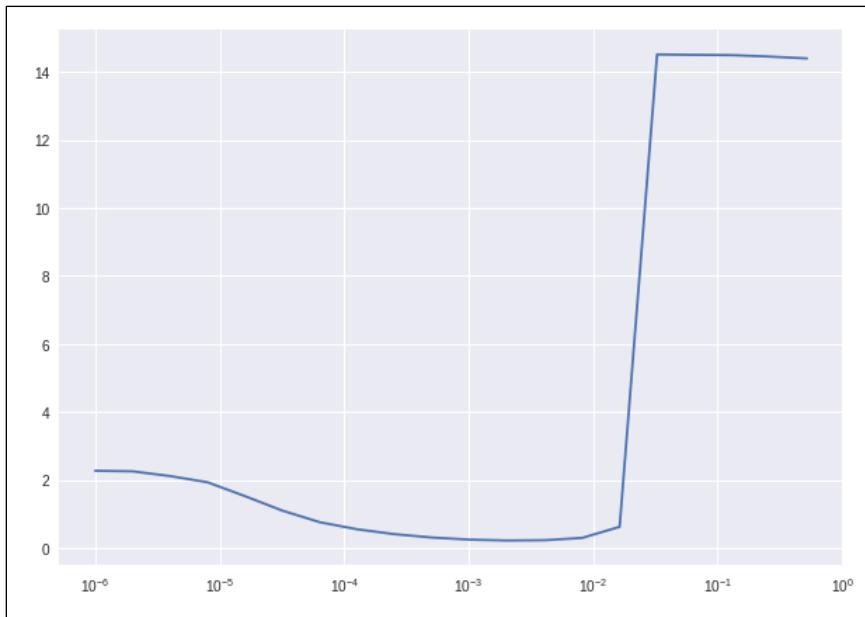
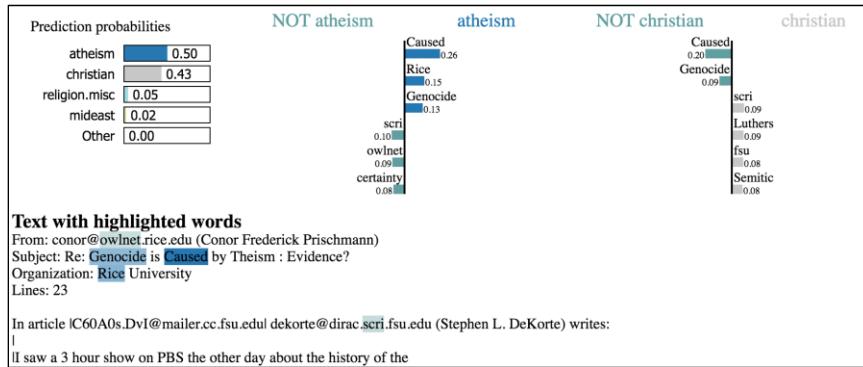


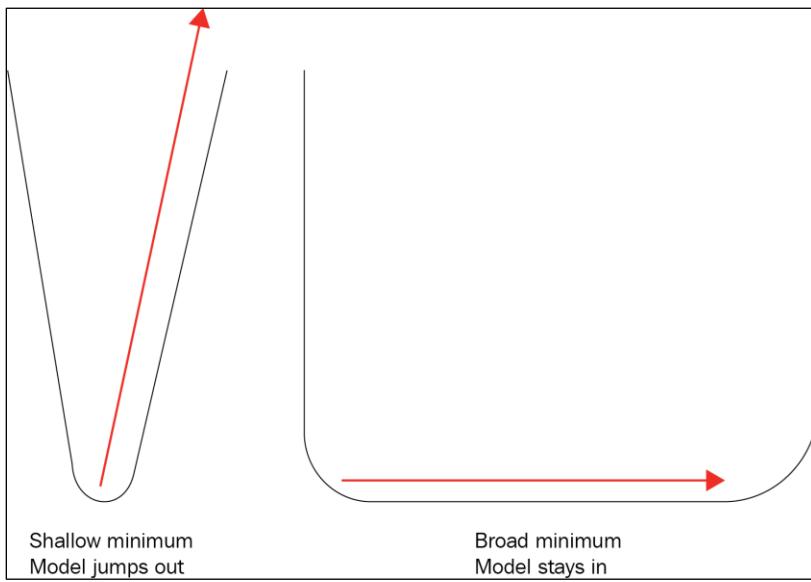
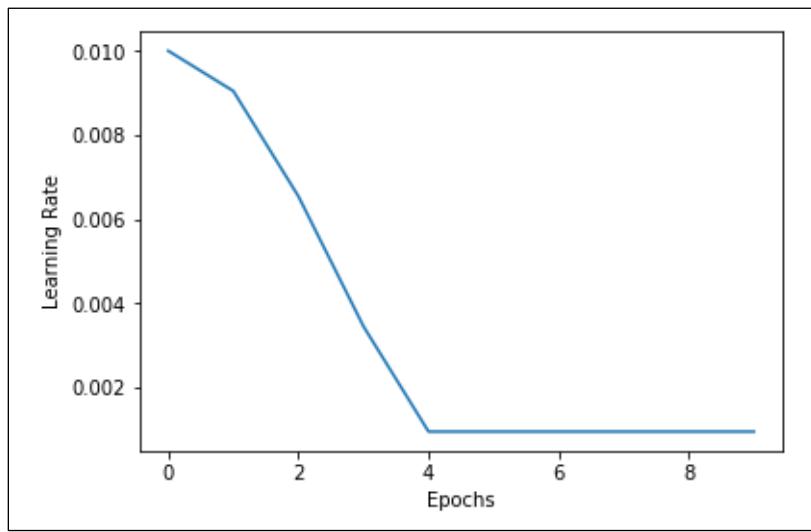


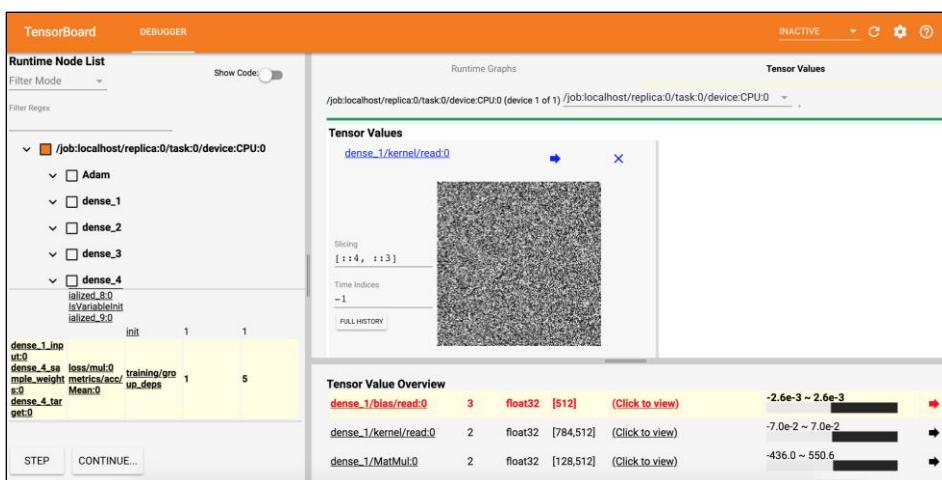
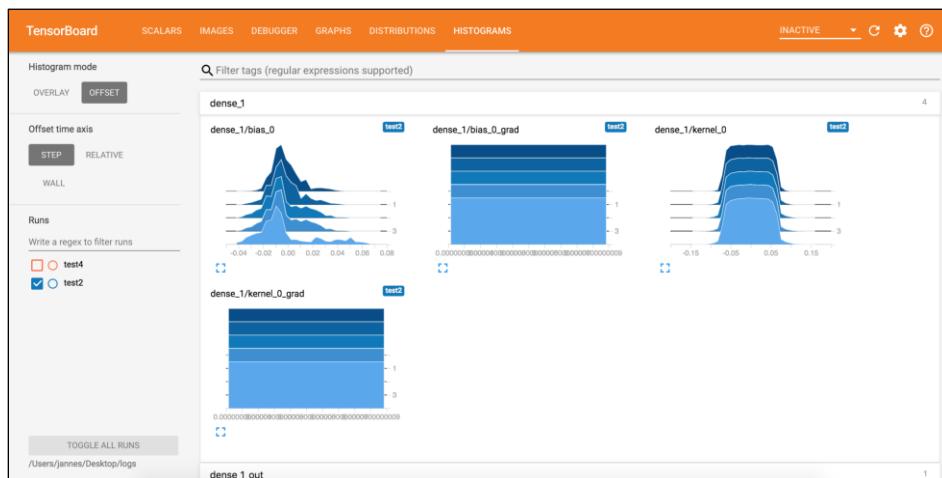
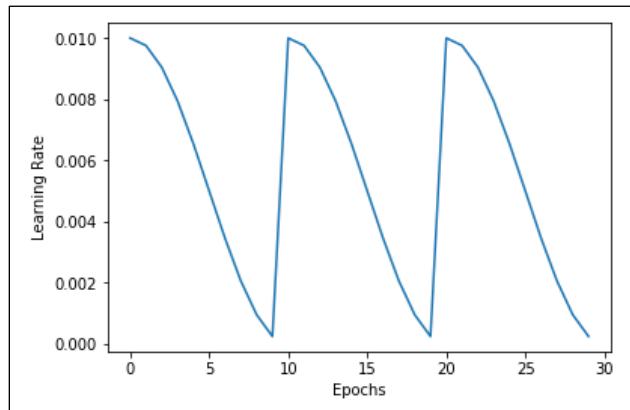


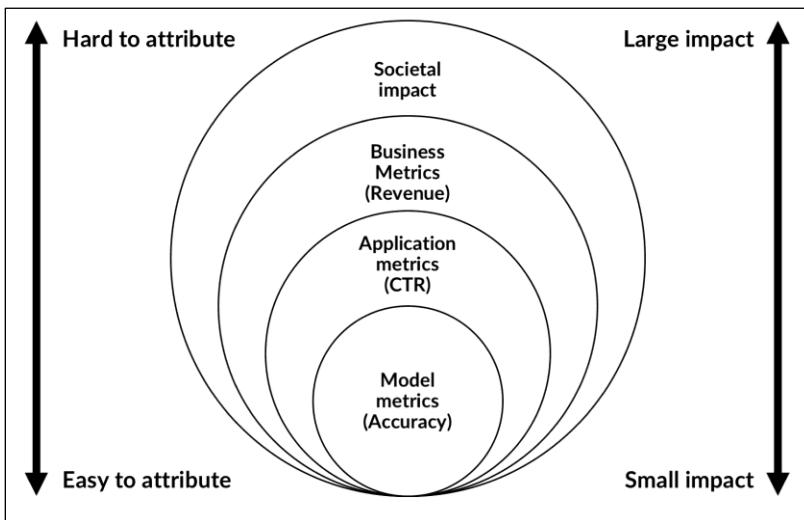
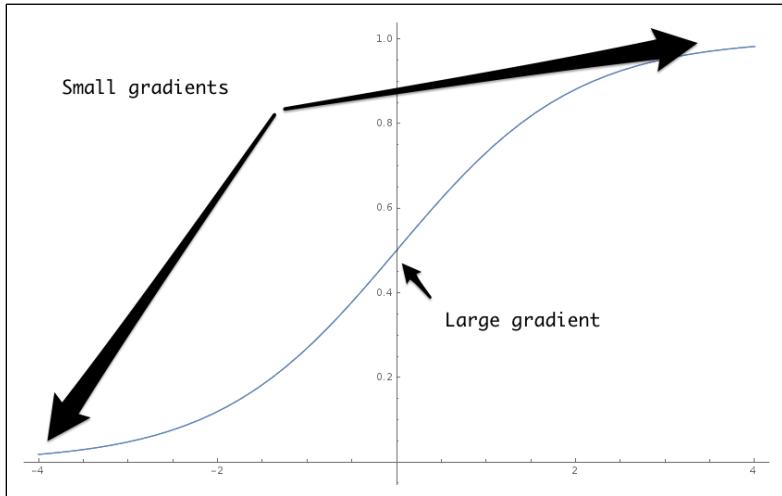


Chapter 8: Privacy, Debugging, and Launching Your Products









Generated by Cython 0.27.2

Yellow lines hint at Python interaction.
Click on a line that starts with a "+" to see the C code that Cython generated for it.

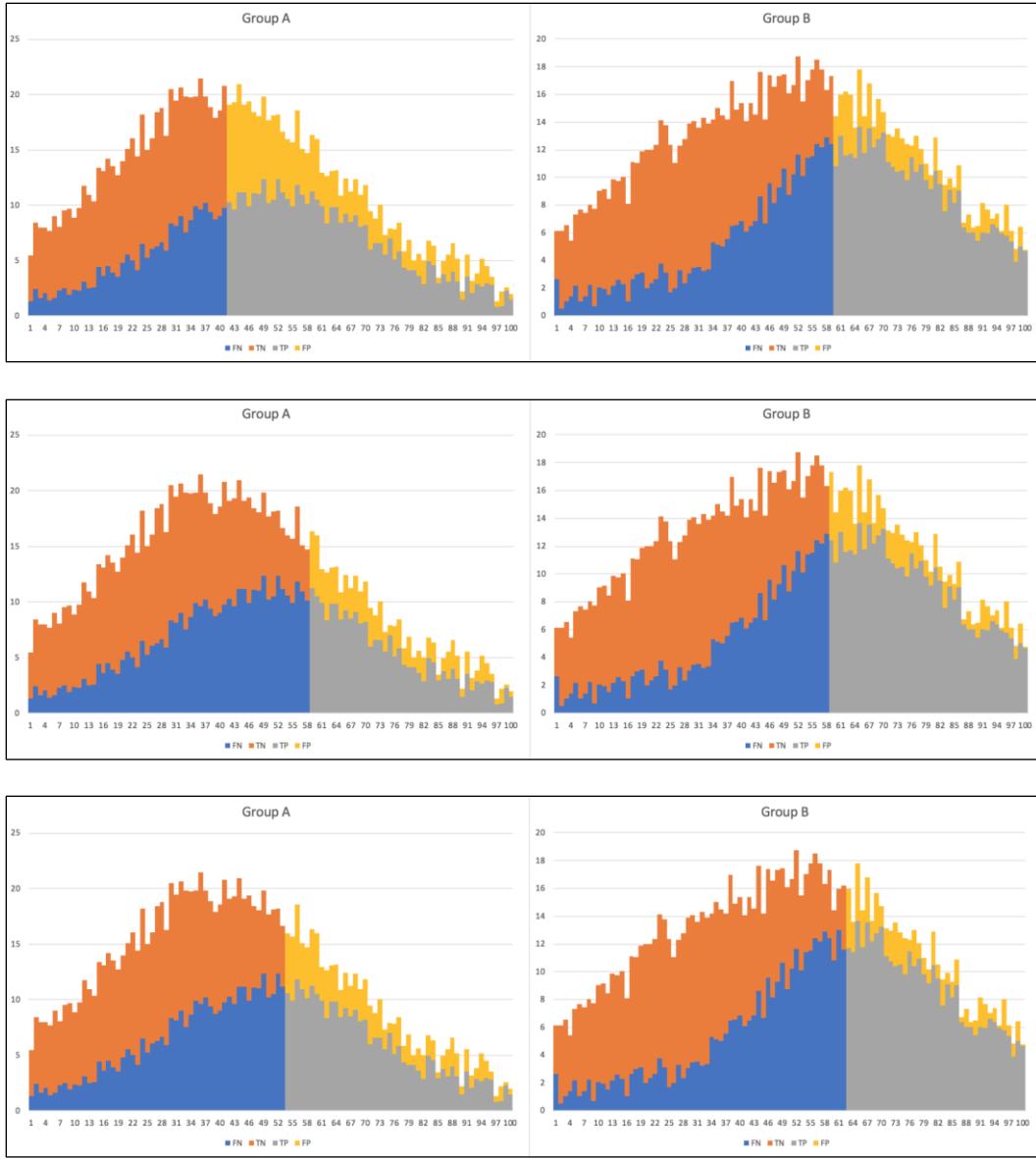
Raw output: [cython_fib_8_5.c](#)

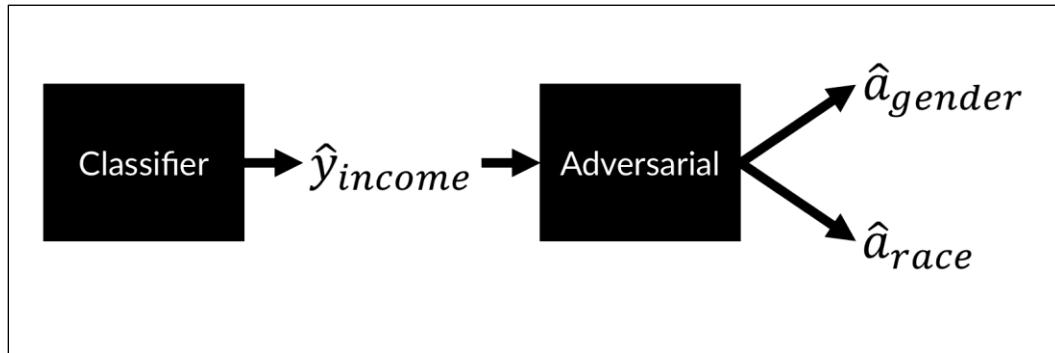
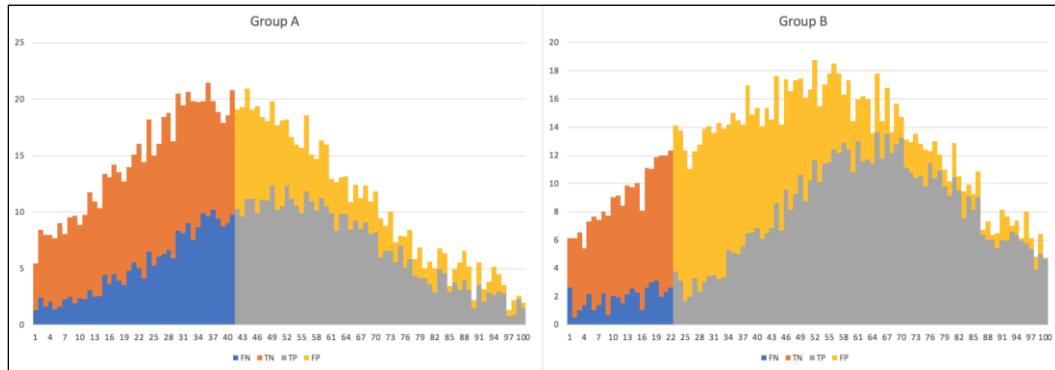
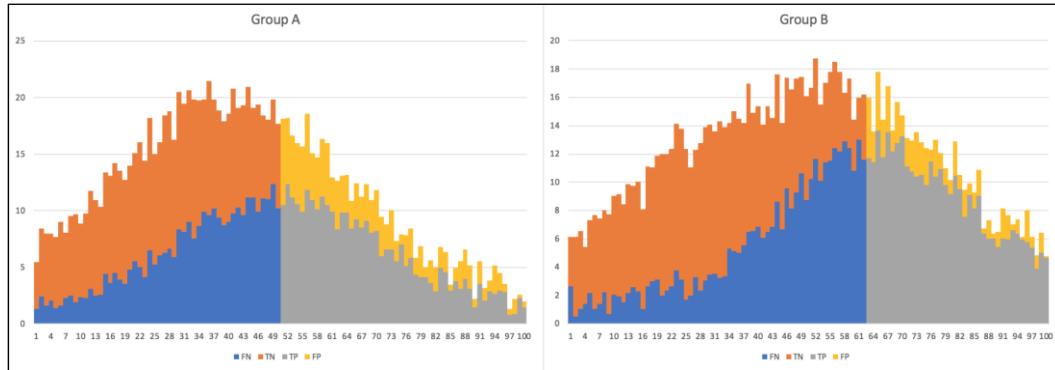
```

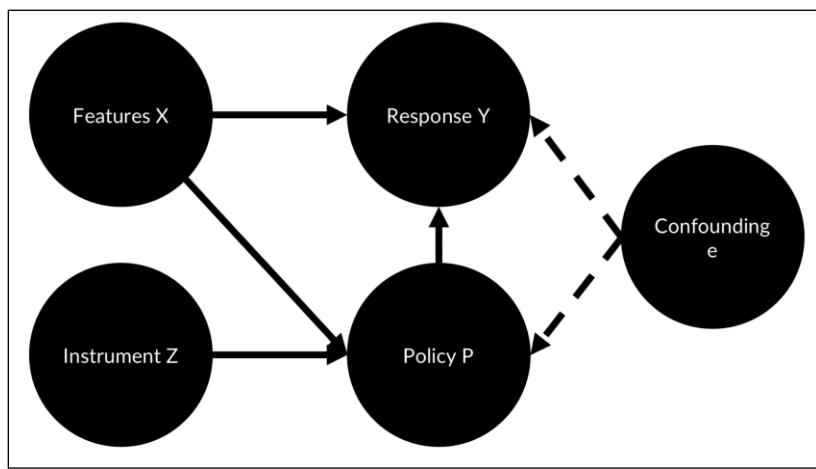
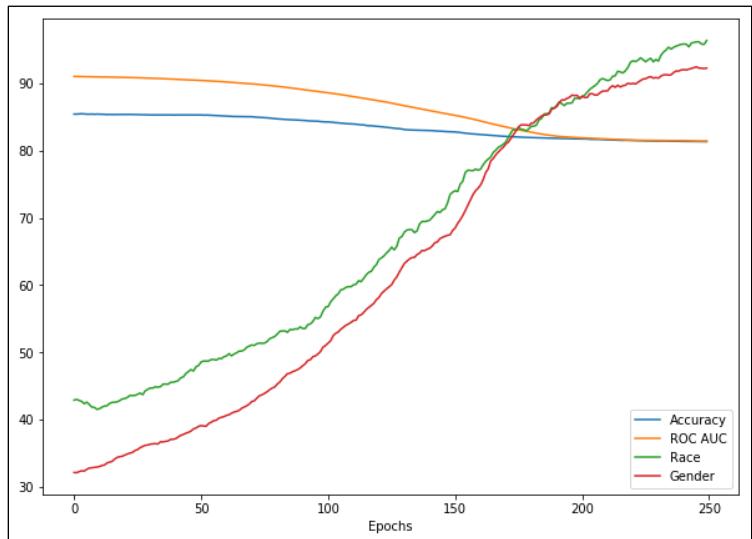
1: from __future__ import print_function
2:
+3: def fib(n):
4:     """Print the Fibonacci series up to n."""
5:     a, b = 0, 1
6:     while b < n:
7:         print(b, end=' ')
8:         a, b = b, a + b
9:     print()

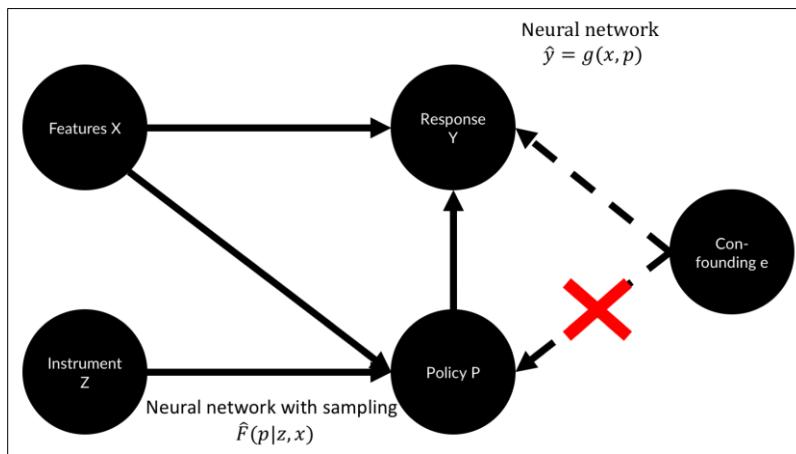
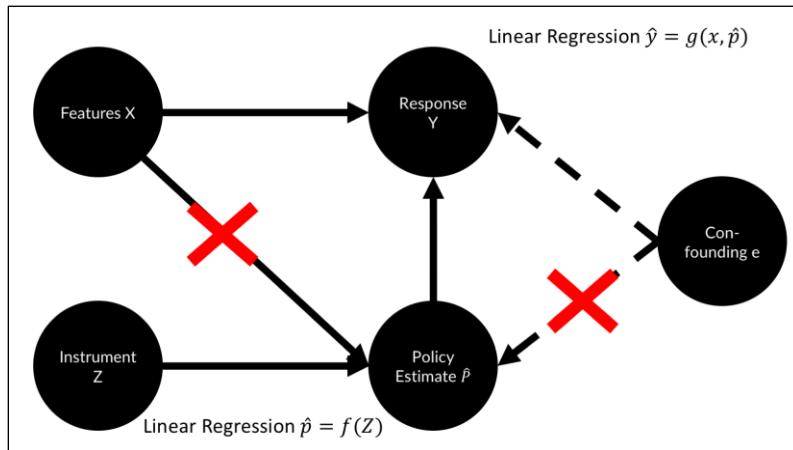
```

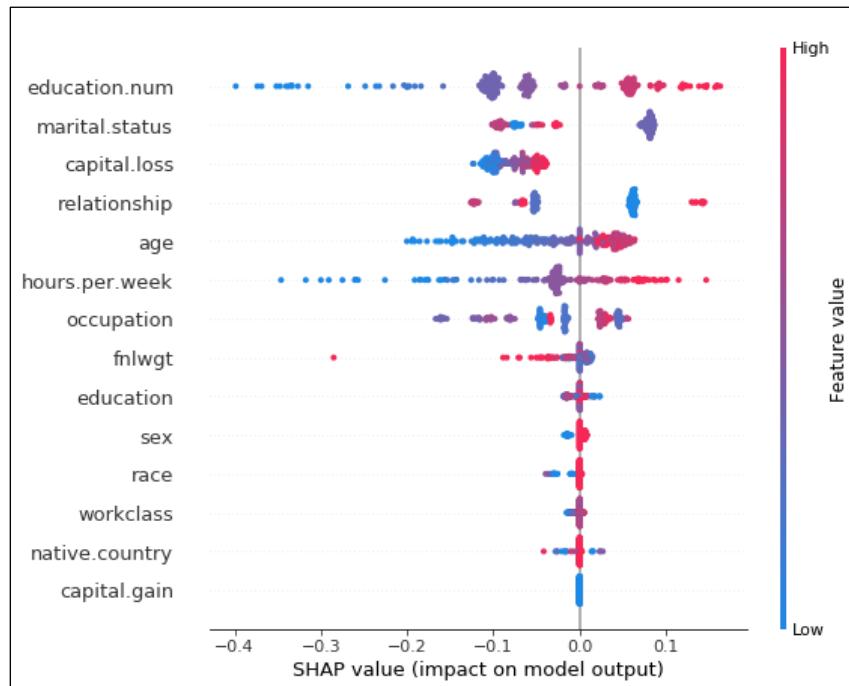
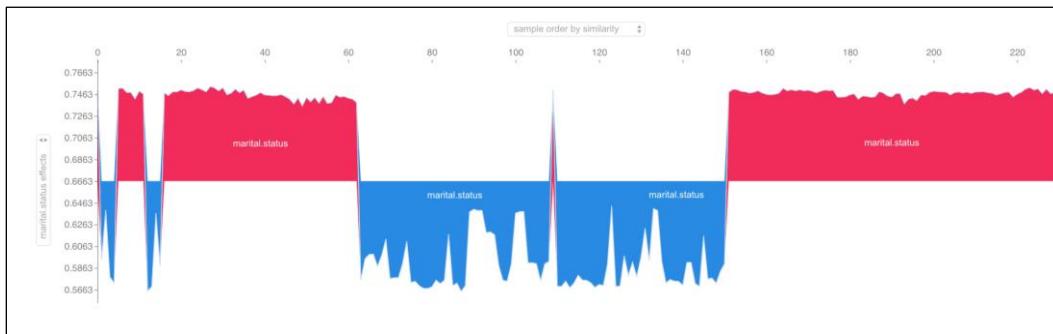
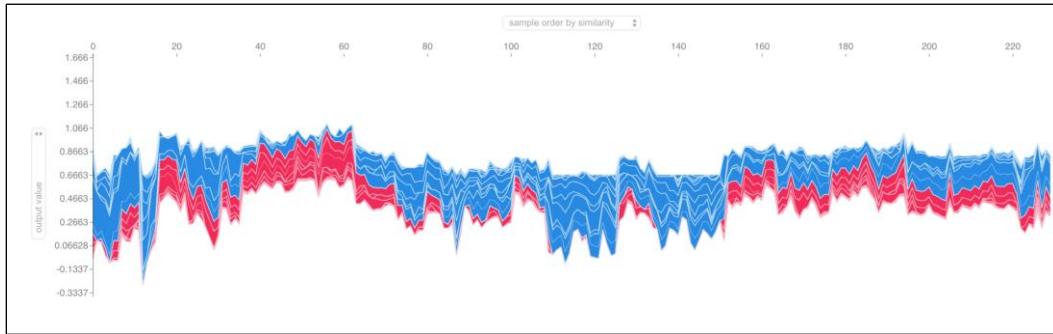
Chapter 9: Fighting Bias

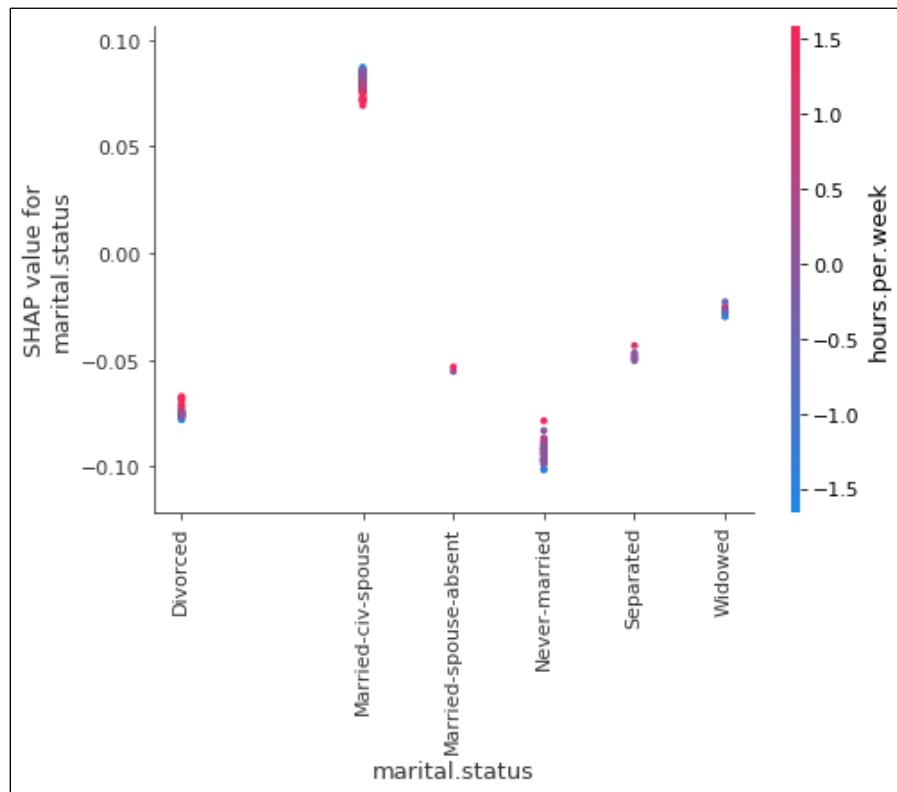












Chapter 10: Bayesian Inference and Probabilistic Programming

