Київський національний університет імені Тараса Шевченка
Факультет радіофізики, електроніки та комп’ютерних систем
Кафедра нанофізики та наноелектроніки

Звіт з лабораторної роботи 2
З курсу «комп’ютерна фізика»

Студента 2 курсу магістратури
кафедра НФНЕ
Кухельного Кирила

київ - 2017

Вступ **3**

Мета **3**

Експериментальна частина роботи **4**

Частина 1 4

Частина 2 4

Частина 1 **5**

Частина 2 **16**

Вступ

Машинне навчання – це спрощена версія процесу навчання, яке відбувається з людиною. Як правило, в машинному навчанні наявний певний набір прикладів, спостережень, реакцій до цих спостережень. Задача полягає у тому, щоб сконструювати такі моделі, які будуть максимально ефективно описувати наявні дані і робити достовірні прогнози. Такими прогнозами можуть бути відповіді на питання: чи є котик на зображенні? Чи є сенс зараз купляти акції певної компанії? Це позитивний чи негативний відгук на товар в Інтернеті? Існує безліч сфер застосування машинного навчання. Така розповсюдженість стала наслідком розвитку Інтернету і накопичення величезної кількості даних, а також тим, що широкому загалу стали доступні великі обчислювальні потужності.

Мета

практично ознайомитися з модельними штучними нейронними мережами, оцінити характеристики ефективності мереж та отримати навички вибору структури мереж для різних типів задач
розпізнавання та аналізу даних.

Експериментальна частина роботи

Частина 1

1. Побудувати одношарову мережу(2 входу, 1 вихід, 1 шар, 1 нейрон у першому шарі), яка виконуватиме логічну операцію XOR і AND.
2. Побудувати двошарову мережу(2 входу, 1 вихід, 1 шар, 1 нейрон у першому шарі), яка виконуватиме логічну операцію XOR і AND.
3. Повторити для 4-5 шарів
4. Повторити модулювання починаючи з одношарової мережі(3 входи, 1 вихід, 1 шар, 1 нейрон у першому шарі) на один з виходів якої подавати постійне значення 1.
5. Порівняти час навчання і похибку для різних структур мереж

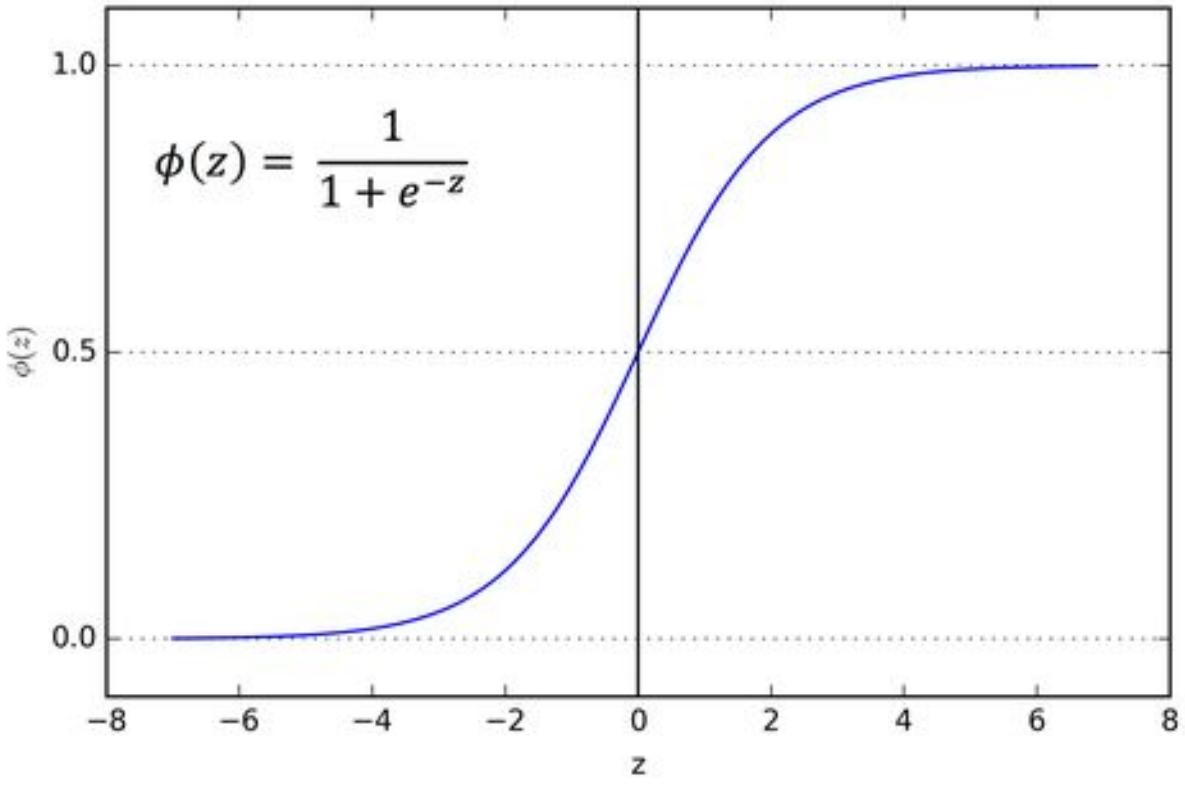
Частина 2

1. Побудувати багатошарову мережу ЦАП(8 входів, 1 вихід, 8 нейронів у 1 шарі, 1 нейрон в останньому)
2. Створити навчальну множину вхідних значень у вигляді ступенів (0 - 7) двійки.
3. Розпізнати 10 чисел, рівномірно розподілених у діапазоні від 0 до 256.
4. повторити вищесказане для багатошарових мереж з прихованими шарами 0, 1, 2, 3, кількість нейронів у кожному шарі розрахувати за формулою варіанту.
5. Порівняти час навчання і похибку для різних структур мереж

Частина 1

Для виконання лабораторної роботи була вибрана мова програмування python та високорівневий фреймворк keras, на базі низькорівневого фреймворка Tensorflow.

Для задач класифікації в основному використовують функцію активації softmax, але враховуючи що в нас данні не в матричному вигляді, вибір було зроблено на користь функції sigmoid. Яка має такий вигляд. $f(x) = sigmoid(x) = \frac{1}{1+e^{-x}}$



Застосуємо бінарну кросентропію **binary_crossentropy**. Вона застосовується завжди, коли ми маємо задачу класифікації між декількома класами. Бінарна кросентропія мінімізується, коли відклик нейромережі максимально чітко наближається до правильного класу. Якщо мітка 1 – бінарна кросентропія буде меншою при активації 0,9, ніж при 0,5.

Використаємо оптимізатор **adam** – метод адаптивного зменшення learning rate (величина кроку, на який ми змінюємо ваги при кожній із ітерацій. Якщо він буде надто великий – нейромережа буде постійно перестрибувати потрібні локальні мінімуми і фактично не навчиться. Якщо буде низьким – нейромережа може застрягнути в одному локальному мінімумі і теж не навчиться). Загалом існує багато методів оптимізацій для підбору кроку, щоб з одного боку підібрати набір вагів, щоб не застрягнути в локальному мінімумі, але щоб і нейромережу навчати швидко. adam поступово то збільшує, то зменшує крок залежно від величини зміни помилки. Якщо система занадто сильно застрягає в мінімумі, метод збільшує крок, потім поступово зменшує, щоб знайти більш глибокий мінімум.
Величина помилки виражається в кросентропії, це негативна логарифмічна вірогідність та залишкова сума квадратів для класифікації та регресії відповідно.Тоді, звичайно, основною метою моделі навчання є зменшення (мінімізації) значення функції втрат по відношенню до параметрів моделі шляхом зміни значень вагових векторів за допомогою різних методів оптимізації, таких як зворотне поширення помилки.

XOR, 2 входи, 1 шар, 1 нейрон, 8000 епох			
Функція активації	Помилка	Точність	Данні
linear	loss: 8.0590	acc: 0.2500	[0, 0] >> -0.0041 [0, 1] >> 1.0491 [1, 0] >> 1.0044 [1, 1] >> 2.0576
sigmoid	loss: 0.693	acc: 0.25	[0, 0] >> 0.5 [0, 1] >> 0.5 [1, 0] >> 0.5 [1, 1] >> 0.5

XOR, 2 входи, 2 шари, (2-1), 8000 епох			
Функція активації	Помилка	Точність	Данні
linear	loss: 8.015	acc: 0.25	[0, 0] >> 0.0 [0, 1] >> 1.703 [1, 0] >> -0.154 [1, 1] >> 1.55
sigmoid	loss: 0.075	acc: 1	[0, 0] >> 0.048 [0, 1] >> 0.927 [1, 0] >> 0.925 [1, 1] >> 0.094

XOR, 2 входи, 2 шари, (3-1), 8000 епох				
Функція активації	Помилка	Точність	час	Данні
linear	loss: 4.03	acc: 0.75	7ms/step	[0, 0] >> -0.009 [0, 1] >> -1.163 [1, 0] >> 1.005 [1, 1] >> -0.148
sigmoid	loss: 0.034	acc: 1	8ms/step	[0, 0] >> 0.004 [0, 1] >> 0.963 [1, 0] >> 0.969 [1, 1] >> 0.061

XOR, 2 входи, 2 шари, (4-1), 8000 епох			
Функція активації	Помилка	Точність	Данні
linear	loss: 8.059	acc: 0.25	[0, 0] >> 0.0 [0, 1] >> -0.663 [1, 0] >> -0.363 [1, 1] >> -1.026
sigmoid	loss: 0.018	acc: 1	[0, 0] >> 0.003 [0, 1] >> 0.995 [1, 0] >> 0.969 [1, 1] >> 0.033

XOR, 2 входи, 2 шари, (5-1), 8000 епох				
Функція активації	Помилка	Точність	час	Данні
linear	loss: 8.059	acc: 0.25	3	[0, 0] >> -0.111 [0, 1] >> -1.142 [1, 0] >> 1.025 [1, 1] >> -0.006
sigmoid	loss: 0	acc: 1	4	[0, 0] >> 0.0 [0, 1] >> 1.0 [1, 0] >> 1.0 [1, 1] >> 0.0

Для операціїї XOR, з додатковим третім входом з постійною 1

XOR, 3 входи, 1 шар, 1 нейрон, 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 4.03	acc: 0.75	7.597	[0 0 1] >> -0.007 [0 1 1] >> 1.001 [1 0 1] >> -1.013 [1 1 1] >> -0.005
sigmoid	loss: 0.693	acc: 0.5	7.613	[0 0 1] >> 0.5 [0 1 1] >> 0.5 [1 0 1] >> 0.5 [1 1 1] >> 0.5

XOR, 3 входи, 2 шари, (2-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 4.03	acc: 0.75	8.364	[0 0 1] >> -0.089 [0 1 1] >> 1.002 [1 0 1] >> -1.116 [1 1 1] >> -0.025
sigmoid	loss: 0.48	acc: 0.75	8.748	[0 0 1] >> 0.003 [0 1 1] >> 0.665 [1 0 1] >> 0.666 [1 1 1] >> 0.667

XOR, 3 входи, 2 шари, (3-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 4.03	acc: 0.75	8.558	[0 0 1] >> -0.012 [0 1 1] >> -1.044 [1 0 1] >> 1.014 [1 1 1] >> -0.018
sigmoid	loss: 0.035	acc: 0.1	9.861	[0 0 1] >> 0.007 [0 1 1] >> 0.964 [1 0 1] >> 0.968 [1 1 1] >> 0.062

XOR, 3 входи, 2 шари, (4-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 3.986	acc: 0.75	8.47	[0 0 1] >> -0.015 [0 1 1] >> 1.003 [1 0 1] >> 1.185 [1 1 1] >> 2.203
sigmoid	loss: 0.004	acc: 1	8.782	[0 0 1] >> 0.004 [0 1 1] >> 0.997 [1 0 1] >> 0.995 [1 1 1] >> 0.004

XOR, 3 входи, 2 шари, (5-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 3.986	acc: 0.75	8.169	[0 0 1] >> -0.014 [0 1 1] >> 1.011 [1 0 1] >> 1.011 [1 1 1] >> 2.035
sigmoid	loss: 0.002	acc: 1	8.378	[0 0 1] >> 0.0 [0 1 1] >> 0.998 [1 0 1] >> 0.998 [1 1 1] >> 0.003

Для операції OR

OR, 2 входи, 1 шар, (1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 12.089	acc: 0.25	7.836	[0 0] >> 0.0 [0 1] >> -0.917 [1 0] >> -1.006 [1 1] >> -1.923
sigmoid	loss: 0.034	acc: 1	7.894	[0 0] >> 0.068 [0 1] >> 0.961 [1 0] >> 0.975 [1 1] >> 1.0

OR, 2 входи, 2 шари, (2-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 0.0	acc: 0.75	8.528	[0 0] >> -0.008 [0 1] >> 1.004 [1 0] >> 1.006 [1 1] >> 2.017
sigmoid	loss: 0.003	acc: 1	8.838	[0 0] >> 0.008 [0 1] >> 0.998 [1 0] >> 0.998 [1 1] >> 0.999

OR, 2 входи, 2 шари, (3-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 4.03	acc: 0.75	9.296	[0 0] >> -0.025 [0 1] >> 1.119 [1 0] >> -0.129 [1 1] >> 1.014
sigmoid	loss: 0.002	acc: 1	8.579	[0 0] >> 0.003 [0 1] >> 0.998 [1 0] >> 0.998 [1 1] >> 0.999

OR, 2 входи, 2 шари, (4-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 12.089	acc: 0.25	8.205	[0 0] >> 0.0 [0 1] >> -0.53 [1 0] >> -0.013 [1 1] >> -0.543
sigmoid	loss: 0.001	acc: 1	8.465	[0 0] >> 0.001 [0 1] >> 0.999 [1 0] >> 0.999 [1 1] >> 1.0

OR, 2 входи, 2 шари, (5-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 0.0	acc: 0.75	8.366	[0 0] >> -0.013 [0 1] >> 1.009 [1 0] >> 1.005 [1 1] >> 2.027
sigmoid	loss: 0.0	acc: 1	9.534	[0 0] >> 0.001 [0 1] >> 1.0 [1 0] >> 1.0 [1 1] >> 1.0

Для операціїї OR, з третім входом

OR, 3 входи, 1 шар, (1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 0.0	acc: 0.75	7.587	[0 0 1] >> -0.001 [0 1 1] >> 1.009 [1 0 1] >> 1.002 [1 1 1] >> 2.012
sigmoid	loss: 0.03	acc: 1	7.534	[0 0 1] >> 0.059 [0 1 1] >> 0.964 [1 0 1] >> 0.978 [1 1 1] >> 1.0
hardsigmoid	loss: 0.0	acc: 1	8.038	[0 0 1] >> 0.0 [0 1 1] >> 1.0 [1 0 1] >> 1.0 [1 1 1] >> 1.0

OR, 3 входи, 2 шари, (2-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 8.059	acc: 0.75	8.6	[0 0 1] >> -0.01 [0 1 1] >> 1.014 [1 0 1] >> -1.395 [1 1 1] >> -0.371
sigmoid	loss: 0.002	acc: 1	8.7	[0 0 1] >> 0.004 [0 1 1] >> 0.998 [1 0 1] >> 0.998 [1 1 1] >> 0.999
hardsigmoid	loss: 0.0	acc: 1	9.9	[0 0 1] >> 0.0 [0 1 1] >> 1.0 [1 0 1] >> 1.0 [1 1 1] >> 1.0

OR, 3 входи, 2 шари, (3-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 4.03	acc: 0.5	8.2	[0 0 1] >> -0.033 [0 1 1] >> 2.212 [1 0 1] >> -1.242 [1 1 1] >> 1.003
sigmoid	loss: 0.002	acc: 1	9.32	[0 0 1] >> 0.004 [0 1 1] >> 0.998 [1 0 1] >> 0.998 [1 1 1] >> 0.999
hardsigmoid	loss: 0.0	acc: 1	9.8	[0 0 1] >> 0.0 [0 1 1] >> 1.0 [1 0 1] >> 1.0 [1 1 1] >> 1.0

OR, 3 входи, 2 шари, (4-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 0.0	acc: 0.5	9.6	[0 0 1] >> -0.507 [0 1 1] >> 1.01 [1 0 1] >> 1.649 [1 1 1] >> 3.167
sigmoid	loss: 0.002	acc: 1	9.4	[0 0 1] >> 0.002 [0 1 1] >> 1.0 [1 0 1] >> 1.0 [1 1 1] >> 1.0
hardsigmoid	loss: 0.0	acc: 1	9.8	[0 0 1] >> 0.0 [0 1 1] >> 1.0 [1 0 1] >> 1.0 [1 1 1] >> 1.0

OR, 3 входи, 2 шари, (5-1), 10000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 0.0	acc: 0.75	8.9	[0 0 1] >> -0.018 [0 1 1] >> 1.018 [1 0 1] >> 1.025 [1 1 1] >> 2.061
sigmoid	loss: 0.0	acc: 1	8.8	[0 0 1] >> 0.001 [0 1 1] >> 1.0 [1 0 1] >> 1.0 [1 1 1] >> 1.0
hardsigmoid	loss: 0.0	acc: 1	9.9	[0 0 1] >> 0.0 [0 1 1] >> 1.0 [1 0 1] >> 1.0 [1 1 1] >> 1.0

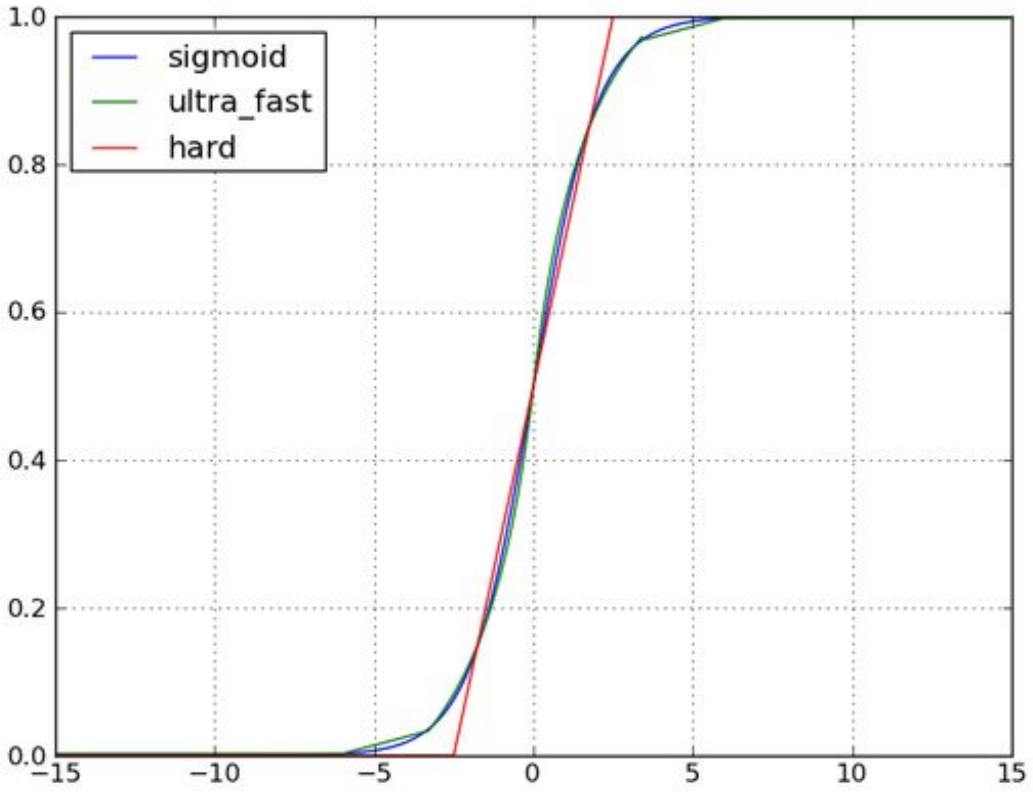
Для підтвердження гіпотези що лінійна функція не може справитись з цією задачою проведемо експеримент і додамо 15 нейронів в перший шар і проведемо 1 мільйон епох навчання.

OR, 2 входи, 2 шари, (15-1), 1’000’000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
linear	loss: 0.0	acc: 0.75	809.357	[0 0] >> -0.012 [0 1] >> 1.002 [1 0] >> 1.003 [1 1] >> 2.017
sigmoid	loss: 0.0	acc: 1	919.534	[0 0] >> 0.0 [0 1] >> 1.0 [1 0] >> 1.0 [1 1] >> 1.0

Висновок:

Функція активації є важливим елементом під час навчання нейронної мережі. І її вибір сильно впливає на отримані результати. Лінійна функція не може адекватно зробити переведення простору через те що область її значень виходить хамежі нашого просторурезультатів. Натомість сігмоїдальні функція чудово справляється з данною задачою, а хад сігмоїд функція за рахунок різких переходів оптимізує процес навчання і дає кращі результати на меншій кількості нейронів ніж звичайна сігмоїдальна. Сігмоїдальним функціям достатньо 3-4 тисяч епох для гарного підбору вагових коефіцієнтів. Для випадку 1 нейрона операція XOR не може бути виконана, тому що перехресне переворення простору неможливо зробити на 1 нейроні. Так як лінійна модель не може її засвоїти через неможливість лінійного розділення просторових ознак. А багатошарові мережі можуть це зробити через те, що роблять неявне перетворення простору ознак з можливістю розділення або ж (в залежності від реалізації) роблять нелінійне розбиття цього простору. Ми реалізували перший варіант - створили двошаровий перцептрон з нелінійної активацією шарів. Перший шар робив нелінійне перетворення, а другий шар - це практично лінійна

регресія, яка працює на перетворення простору ознак. Зростання часу навчання мережі відносно кількості шарів і нейронів у них присутнє. Проте данних не достатньо щоб сказати чи воно лінійне чи експоненційне.



Частина 2

Для навчання ЦАП. була взята тестова вибірка
[0, 0, 0, 0, 0, 0, 0, 1],
[0, 0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0],

На тренувальній вибірці були отримані чудові результи

```
[0 0 0 0 0 0 0 1]  >>  1.0      1.0
[0 0 0 0 0 0 1 0]  >>  2.0      2.0
[0 0 0 0 0 1 0 0]  >>  4.0      4.0
[0 0 0 1 0 0 0 0]  >>  16.0     16.0
[0 0 1 0 0 0 0 0]  >>  32.0     32.0
[0 1 0 0 0 0 0 0]  >>  64.0     64.0
[1 0 0 0 0 0 0 0]  >>  128.0    128.0
```

Проте на тестувальних данних вони суттєво відрізнялись.
Данні наведено для тестувальної вибірки, так як з тренувальною вседобре.

ЦАП, 8 входів, 2 шари, (8-1), 12’000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
sigmoid	loss: 1.037	acc: 0.0	17.504	[0 0 0 0 0 1 1 0] >> 6.0 1.057 [0 0 0 0 1 0 0 1] >> 9.0 0.816 [0 0 0 0 1 0 1 1] >> 11.0 0.563 [0 0 0 1 1 0 0 0] >> 24.0 8.119 [0 1 1 0 0 1 0 0] >> 100.0 38.052 [0 1 1 1 0 0 0 0] >> 112.0 88.351 [1 0 0 1 0 1 1 0] >> 150.0 7.569

Згідно варіанту будемо почергово додавати шари на 11, 9, 7, 5 нейронів.

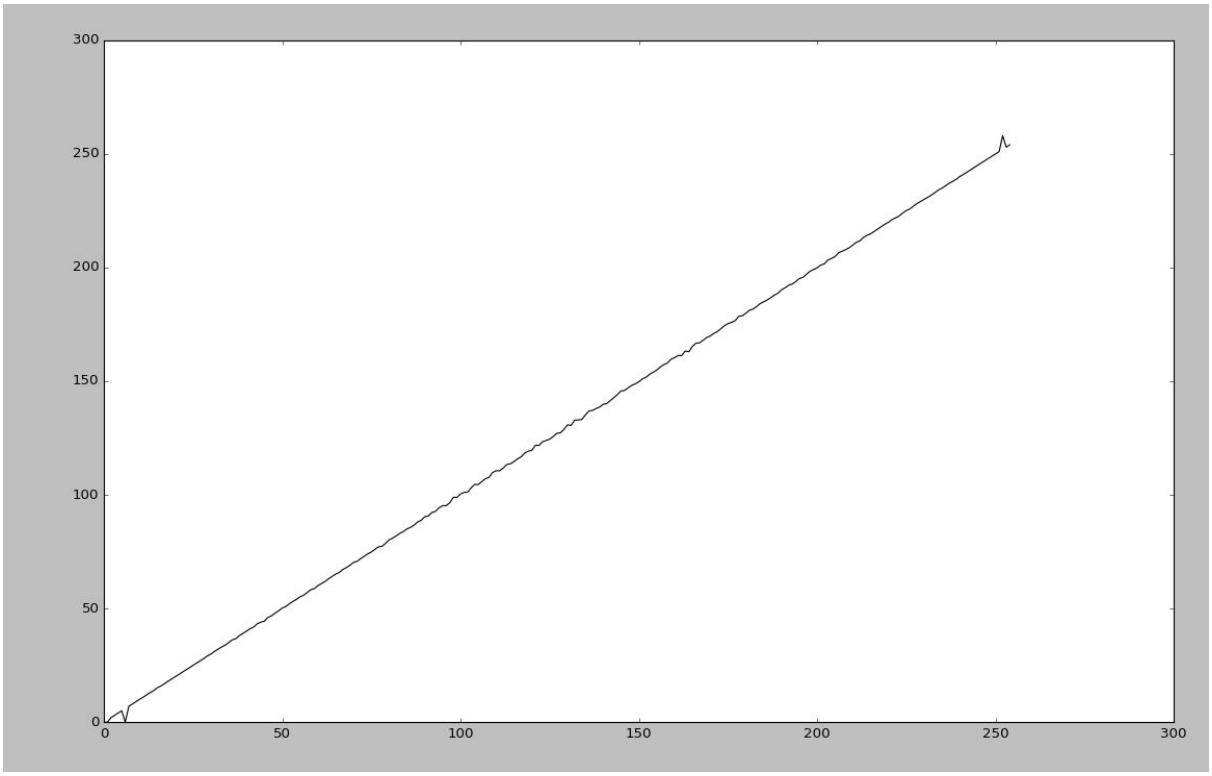
ЦАП, 8 входів, 3 шари, (8-11-1), 12’000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
sigmoid	loss: 1.058	acc: 0.0	17.319	[0 0 0 0 0 1 1 0] >> 6.0 1.04 [0 0 0 0 1 0 0 1] >> 9.0 0.875 [0 0 0 0 1 0 1 1] >> 11.0 0.763 [0 0 0 1 1 0 0 0] >> 24.0 5.341 [0 1 1 0 0 1 0 0] >> 100.0 18.636 [0 1 1 1 0 0 0 0] >> 112.0 51.991 [1 0 0 1 0 1 1 0] >> 150.0 8.141

ЦАП, 8 входів, 3 шари, (8-11-9-1), 12’000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
sigmoid	loss: 1.019	acc: 0.0	19.46	[0 0 0 0 0 1 1 0] >> 6.0 1.13 [0 0 0 0 1 0 0 1] >> 9.0 0.967 [0 0 0 0 1 0 1 1] >> 11.0 0.935 [0 0 0 1 1 0 0 0] >> 24.0 2.634 [0 1 1 0 0 1 0 0] >> 100.0 3.067 [0 1 1 1 0 0 0 0] >> 112.0 9.608 [1 0 0 1 0 1 1 0] >> 150.0 52.623

ЦАП, 8 входів, 3 шари, (8-11-9-7-1), 12’000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
sigmoid	loss: 0.829	acc: 0.0	19.696	[0 0 0 0 0 1 1 0] >> 6.0 1.266 [0 0 0 0 1 0 0 1] >> 9.0 0.997 [0 0 0 0 1 0 1 1] >> 11.0 0.992 [0 0 0 1 1 0 0 0] >> 24.0 3.678 [0 1 1 0 0 1 0 0] >> 100.0 10.138 [0 1 1 1 0 0 0 0] >> 112.0 52.982 [1 0 0 1 0 1 1 0] >> 150.0 105.534

ЦАП, 8 входів, 3 шари, (8-11-9-7-5-1), 12’000 епох				
Функція активації	Помилка	Точність	час, сек	Данні
sigmoid	loss: 0.859	acc: 0.0	20.7	[0 0 0 0 0 1 1 0] >> 6.0 1.703 [0 0 0 0 1 0 0 1] >> 9.0 1.027 [0 0 0 0 1 0 1 1] >> 11.0 1.018 [0 0 0 1 1 0 0 0] >> 24.0 9.753 [0 1 1 0 0 1 0 0] >> 100.0 20.2 [0 1 1 1 0 0 0 0] >> 112.0 69.45 [1 0 0 1 0 1 1 0] >> 150.0 35.514

Для вирішення цього завдання треба збільшити вибірку. Заради експерименту було побудовано мережу з 255 входами. І використана функція активації RELU. Та отримані такі данні. Точність 0.98,що можна вважати ідеальним результатом.



Висновок:

Данна архіректура нейромережі не дозволяє отримати високу точність. Очевидно що кількості данних замало для навчання нейромережі. Тому кількість епох і збільшення шарів не додають точності. Зростання часу також присутнє. Пояснення співпадає з попередніми висновками.