Tab 1

**Machine learning - unit 10**

בקובץ הזה אני בעיקר אכתוב פירושים ודברים שימושיים אשר הוצאתי מהן.

importent concepts

**Important concepts\definitions**

# **Feature vector -**

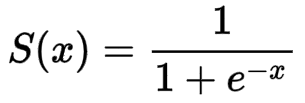
A vector that represents a list of data that represents non numerical data.

# **R\RN\RN\*M -**

R represents a set of real numbers. When it's into the power of a number it specifies the structure of the data that makes up the set.

In other words, R is a set of single real numbers, RN is a set of N length vectors of real numbers, and RN\*M is a set of n\*m matrixes (M = number of columns) of real numbers.

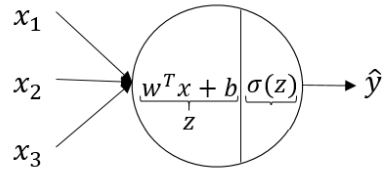
**Sigmoid -**

****

This is the sigmoid function. It is used to translate all numbers to a number between 0 and 1. It's useful when doing logical regression, where the answer is either 0 or 1. Another important note is that the derivative of sigmoid contains only sigmoid. This makes it easy to derive, and allows for one singular calculation of sigmoid, which also gives the derivative.

e is Euler's number - a mathematical constant(about 2.7). The derivative of ex is ex, which is a very powerful tool.

**Perceptron -**

****

This is a Perceptron, you can think of this as the basic calculation unit, whose task is to determine something(ŷ). The first part does the calculation and the second translates it into a value between 0 and 1 (0 is false and 1 is true).

**Cross entropy loss function -**

This is a function that measures the model's success\failure. This kind of function takes into consideration the result and how close the model was to that result. Even if the answer was wrong, a 50-50 mistake will result in a lower mistake rate than a 30-70 one. This concept is also correct when talking about multiple possibilities(is cat\is dog\is horse).

**Binary Cross-Entropy Loss -**

****

**This is an incredibly important function!!**

This is the standard function used for Binary Cross-Entropy Loss. basically a cross entropy loss function with only 2 possible options.

This also includes all types of yes no questions.

The function has 2 sides

usfull python methods

**Useful\important python\numpy methods and tricks**

**Flatten -**

Used to flatten an array into a 1D one. Useful when trying to work with pictures\pixels. Can be used for general feature vector creation.

**Exm - flattened\_array = array.flatten()**

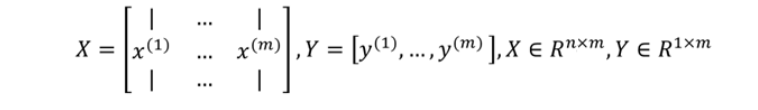
רגרסיה לוגית - 2

**רגרסיה לוגית - 2**



This just means that given the data x + y, x is known to be a vector from the [R^n set](#t2z6k1prue2t), and y is either 0 or 1.

The weird symbol is just a part of a series.



-

This is a bit more complicated - it represents combining the training data into a single big group - the pictures are made into a matrix where each column is one picture(each row is an rgb value), and the y is made into a single row\metrix with one row.

RN\*M = a set of matrixes.

R1\*M = a “metrix” with one row.



This is the stage where the magic happens.

W is a column vector(a vector that is one column and not one row), where each value is the “weight” of a single rgb value of a single pixel. You can think of the weight as the importance of the x value - this is one of the values we will modify. For example, if we train the model using only pictures where an orange cat is in the middle, it will give the rgb-red value in the middle pixels a bigger weight.

B is a singular value that acts as a “stabilizer”. Sense the goal is to make a function that will pass through a dot that represents the truth(a value that will say if the condition is true\not true), we need the stabilizer. Without b the graph will have to go through 0,0, meaning that it will be restricted, and the function won't be accurate (it won't be capable of “choosing” the most efficient line, and will instead choose the most efficient line amongst the ones that go through 0,0).





This is the error function. It's a bit more complicated than last time. Basically, it takes into consideration each option(is a cat\ isn't a cat), by having one expression for each case, when for each option the other expression will equal 0. The lan function is used due to its useful and matching characteristics.

This function is extremely important! It's used in all logical yes\no tasks!