	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

Age and Gender Classification


(Specifications)

Bui Gia Huy^{*†}

March 6, 2019

^{*}Artificial Intelligence Department


[†]Email: huygb@vietnews24.com

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

DOCUMENT CHANGE LOG


Rev.	Description	Author	Date	Approved	Date
0.1	Draft 1	Huy G.B	2019/03/04		

This document has been based on template TL-SSW-101 Revision 2.0

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Scope	4
2	Related Works	5
2.1	Classification algorithms	5
2.1.1	Perceptron	5
2.1.2	Logistic Regression	6
2.1.3	Artificial Neural Network	8
2.2	Web Application	10
2.2.1	Frontend	10
2.2.2	Backend	10
2.3	Example: Dog and Cat Classification	10
3	System Architecure	11
3.1	System Overall	11
3.2	Fronted	11
3.3	Backend	11
3.4	Neural Network	11
4	Plan	12
5	References	13

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

1 Introduction


- This *Capstone Project 1* assists for my learning about Machine learning and Deep learning in training course of D-Soft company that Mr. Trung Anh is my supervisor.
- Through this project, i want to improve some skills about coding, researching relate to machine learning.
- It has a lot of interesting applications about machine learning but in this project i will choose topic **Age and Gender Classification** because D-Soft company where i am working, has a project relate to this problem.
- Basically, my application, which I build, has a website interface that anyone can also access. At that, website allows you submit an image of single people and then it will returns information about age and gender of person inside the picture.

1.1 Purpose

- Reinforce knowledge about machine learning learned from Coursera *Andrew Ng*.
- Build a funny application that apply machine learning to a real world problem.

1.2 Scope

- This project in range of D-Soft's training course.
- The project focus mainly about:
 - + Build a simple web frontend and backend.
 - + Training a machine learning algorithm.

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

2 Related Works

- The system that I build relate to **classification** problem so it is useful to list some machine learning algorithm.
- In addition to ML algorithms serve for classify age and gender of a single people, the system is also needed host on a server for demo session. Thus, we will introduce some concepts relate to *Web Application* such as *Frontend* and *Backend*.

2.1 Classification algorithms

- It has a lot of machine learning algorithms relate to classification problem but here, I only introduce some ones that I learned.

2.1.1 Perceptron


- In machine learning, the **perceptron** is an algorithm for **supervise learning** of **binary classification**. A binary classifier a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class. It a type of **linear classifier**, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.
- In the modern sense, the perceptron is an algorithm for learning a binary classifier called a threshold function: a function that maps its input x (a real-valued vector) to an output value $f(x)$ (a single binary value):

$$f(x) = \begin{cases} 1, & (w \cdot x + b > 0) \\ 0, & otherwise \end{cases}$$

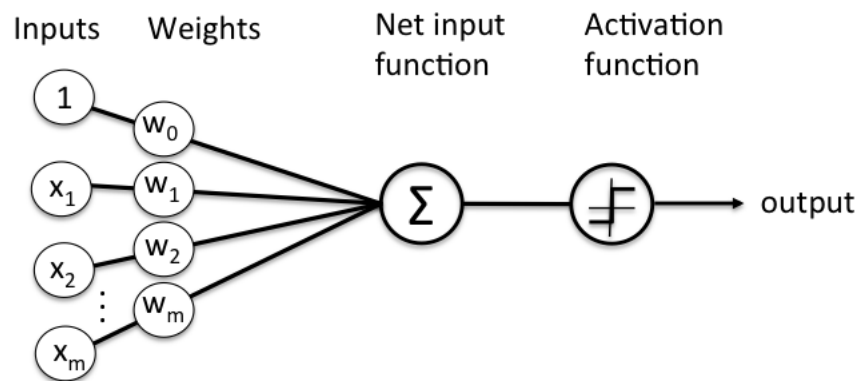
where \mathbf{w} is a vector of real-valued weights, $\mathbf{w} \cdot \mathbf{x}$ is the dot product $\sum_{i=1}^m w_i x_i$, where m is the number of inputs to the perceptron, and b is the bias. The bias shifts the decision boundary away from the origin and does not depend on any input value.

The value of $f(x)$ (0 or 1) is used to classify x as either a positive or a negative instance, in the case of binary classification problem. If b is negative, then the weighted combination of inputs must produce a positive value greater than $|b|$ in order to push the classifier neuron over the 0 threshold. Spatially, the bias alters the position of the decision boundary. The perceptron learning algorithm does not terminate if the learning set is not linear separable. If the vectors are not linearly separable learning will never reach a point where all vectors are classified properly. The most famous example of the perceptron's inability to solve problems with linearly nonseparable vectors is the Boolean exclusive-or problem. The solution spaces of decision boundaries for all binary functions and learning behaviors are studied in the reference.

In the context of neural networks, a perceptron is an artificial neuron using the *Heaviside*

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

step function as the activation function. The perceptron algorithm is also termed the *single layer perceptron*, to distinguish it from *multilayer perceptron*, which is a misnomer for a more complicated neural network. As a linear classifier, the single-layer-perceptron is the simplest *feedforward neural network*.




Schematic of Rosenblatt's perceptron.

(Source: Sebastian Raschka)

2.1.2 Logistic Regression

- **Logistic Regression**, also known as *Logit Regression* is a mathematical model used in statistics to estimate (guess) the probability of an event happens (1) or the event does not happen (0). So given some feature \mathbf{x} , it tries to find out whether some event \mathbf{y} happens or not. So \mathbf{y} can either be 0 or 1. In the case where the event happens, \mathbf{y} is given the value 1. If the event does not happen, then \mathbf{y} is given the value 0. For example, if \mathbf{y} represent whether a sports team wins a match, then \mathbf{y} will be 1 if they win the match or \mathbf{y} will be 0 if they do not. This is known as *Binomial Logistic Regression*. There is also another form of Logistic Regression which uses multiple values for the variable \mathbf{y} . This form of Logistic Regression is known as *Multinomial Logistic Regression*.

- Logistic Regression uses the logistic function to find a model that fits with the data points. The function gives an 'S' shape curve to model the data. The curve is restricted between 0 and 1, so it is easy to apply when \mathbf{y} is binary. Logistic Regression can then model events better than linear regression, as it shows the probability for \mathbf{y} being 1 for a given \mathbf{x} value. Logistic Regression is used in statistics and machine learning to predict values of an input from previous

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

test data.

Logistic regression is an alternative method to use other than the simpler *Linear Regression*. Linear regression tries to predict the data by finding a linear - straight line - equation to model or predict future data points. logistic regression does not look at the relationship between two variables as a straight line. Instead, Logistic regression uses the logarithm function to find the relationship between the variables and uses test data to find the coefficients. The function can then predict the future results using these coefficients in the logistic equation.

Logistic regression uses the concepts of odds ratios to calculate the probability. This is defined as ratio of the odds of an event happening to its not happening. For example, the probability of a sports team to win a certain match might be 0.75. The probability for that team to lose would be $1 - 0.75 = 0.25$. The odds for that team winning would be $0.75/0.25 = 3$. This can be said as the odds of the team winning are 3 to 1.

The odds can be defined as:

$$Odds = \frac{P(y = 1|x)}{1 - P(y = 1|x)}$$

The natural logarithm of the odds ratio is then taken in order to create the logistic equation. The new equation is known as the logit:

$$Logit(P(x)) = \ln \left(\frac{P(y = 1|x)}{(1 - P(y = 1|x))} \right)$$

In logistic regression the Logit of the probability is said to be linear with respect to x, so the logit becomes:

$$Logit(P(x)) = a + bx$$

Using the two equations together then gives the following:

$$\frac{P(y = 1|x)}{1 - P(y = 1|x)} = e^{a+bx}$$

This then leads to the probability:

$$P(y = 1|x) = \frac{e^{a+bx}}{1 + e^{a+bx}} = \frac{1}{1 + e^{-(a+bx)}}$$

The final equation is the logistic curve for Logistic regression. It models the non-linear relationship between x and y with an 'S-like' curve for the probabilities that y = 1, that event the y occurs. In this example **a** and **b** represent the gradients for the logistic function. The logit equation can then be expanded to handle multiple gradients. This gives more freedom with how the logistic curve matches the data. The multiplication of two vectors can then be used to

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

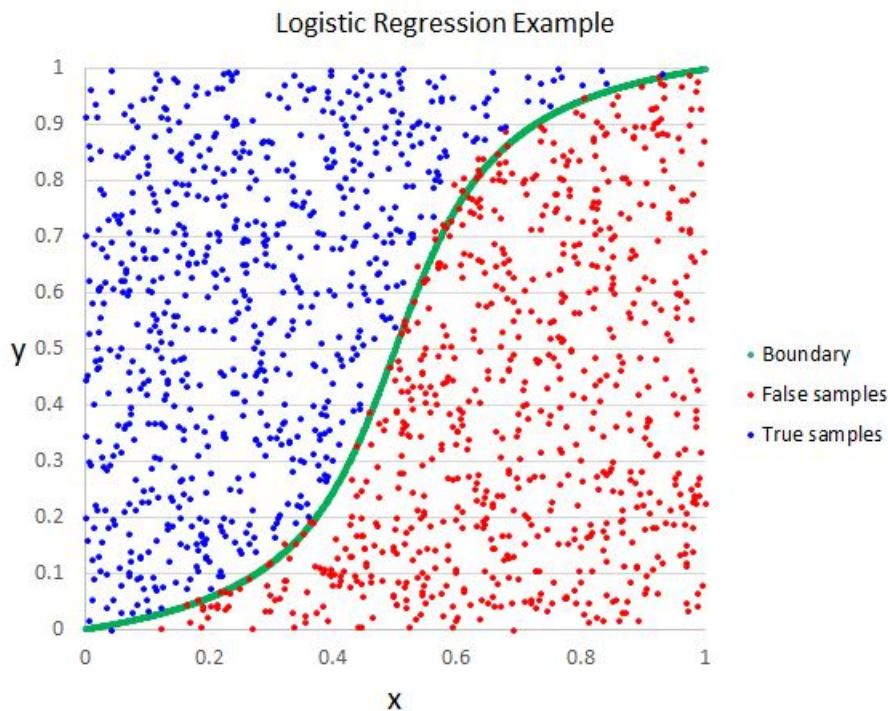
model more gradient values and give the following equation:

$$\text{Logit}(P(x)) = w_0x^0 + w_1x^1 + w_2x^2 + \dots + w_nx^n = w^T x$$

In this equation $w = [w_0, w_1, \dots, w_n]$ and represents the n gradients for the equation. The powers of x are given by the vector $x = [1, x, x^2, \dots, x^n]$. These two vectors give the new logit equation with multiple gradients. The logistic equation then can be changed to show this:

$$P(y = 1|x) = \frac{1}{1 + e^{-(w^T x)}}$$


This is then a more genral logistic equation allowing for more gradient values.



(Source: helloacm.com)

2.1.3 Artificial Neural Network

- **Artificial Neural Networks (ANN)** or **connectionist systems** are computing systems inspired by the *biological neural networks* that constitute animal brains. The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. For example, in *image recognition*, they might learn to identify images that contains cats by analyzing example images that have been manually labelled as "cat" or "no cat" and using the

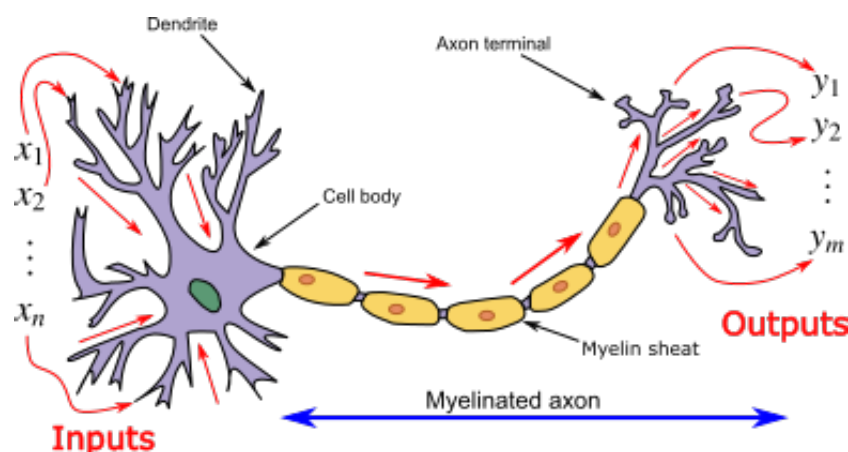
	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

results to identify cats in other images. They do this without any prior knowledge about cats, for examples, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the learning material that they process.

An ANN is based on collection of connected units or nodes called *artificial neurons*, which loosely model the *neurons* in a biological brain. Each connection, like the *synapses* in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected it.


In common ANN implementations, the signal at a connection between artificial neurons is *real number*, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called 'edges'. Artificial neurons and edges typically have a *weight* that adjust as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layer multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. ANNs have been used on variety of tasks, including *computer vision*, *speech recognition*, *machine translation*, *social network filtering*, *playing board and video games* and *medical diagnosis*.

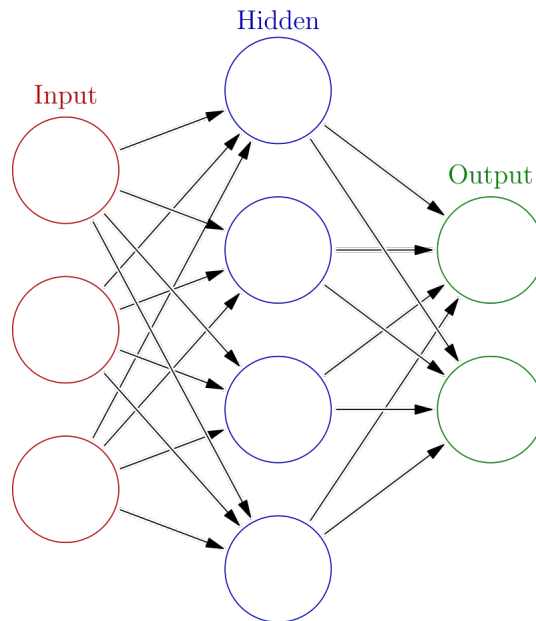


Neuron and myelinated axon, with signal flow from inputs at dendrites to outputs at axon terminals.

(Source: wiki)

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

An ANN is an interconnected group of nodes, similar to the vast network of neurons. Here, each node represents an artificial neuron and arrow represent a connection from the output of one artificial neuron to the input of another.




(Source: wiki)

2.2 Web Application

2.2.1 Frontend

2.2.2 Backend

2.3 Example: Dog and Cat Classification

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn


3 System Architecure

3.1 System Overall

3.2 Fronted


3.3 Backend

3.4 Neural Network

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

4 Plan

Work Breakdown Structure & Estimate			
#	Activities		Expected (man-days)
1	Preparation	1	
	1.1	Setup the development environment	0.5
	1.2	Search + Pre-processing Dataset	0.5
2	Design and Implement	3	
	2.1	Training + Testing model	2
	2.2	Frontend + Backend	1
	Total (man-days)		4

	DSOFT - Vietnam Joint Stock Company		
	Capstone Project 1		http://www.d-soft.com.vn
	Reference:	Revision: 0.1	dsoft@d-soft.com.vn

5 References

1. https://talhassner.github.io/home/publication/2015_CVPR
2. https://talhassner.github.io/home/projects/cnn_agegender/CVPR2015_CNN_AgeGenderEstimate.pdf
3. <https://gilscvblog.com/2015/11/19/age-and-gender-classification-using-deep-convolutional-neural-networks/>
4. <http://dlib.net/>
5. <https://js.tensorflow.org/>