# Udacity Machine Learning Nanodegree
# Capstone Project Proposal

Marcio Nicolau

October 11, 2018

## 1 Domain Background

In general, machine learning can be divided into three large areas designated as supervised, unsupervised and reinforcement learning, each one is recommended to solve a specific kind of problem, and for more complex tasks as learning (agent) by interaction (environment) and try to understand how to get the maximum overall reward during this process. Among all supervised learning techniques, one which gains a considerable attention is Deep Learning, that has been responsible for a dramatical improvement of state-of-the-art applications for areas from speech recognition to natural language process. The deep learning approach to reinforcement learning leads to an improvement of the problems complexity that Markov Decision Process (MDP) could solve.

Some of these recent achievements are extensively noticed, like the DeepMind AlphaGo and OpenAI Dota2. A good source of information related to this reinforcement learning context and applications could be found in the Sutton and Barto book "Reinforcement Learning: An introduction" (2th Ed., 2017). Alongside the game's achievements, reinforcement learning is also widely used in the robotics field, to train robots to accomplish tasks without human interaction or domain-specific knowledge.

My personal motivation to work with reinforcement learning is related to better understand the process and application of this area of machine learning to a real-world problem. This is the closest implementation available today and related to the Artificial General Intelligence (AGI) concepts.

## 2 Problem Statement

In general, a person who want to loose weight or pretend to follow a well-defined and health meal plan, certainly well make an appointment with a nutritionist or dietician. This professional works the meals planning a recommendation for this individual, after collect some basic information about routines, food intake habits, goals, sleep pattern, activities (work, gym, etc.) and food preferences and restrictions. This approach works very well and for a long time, but, in some occasions, the person who have been following the meal plan need to make a change or adjust, based on eventual situations and this could cause a problem.

This proposal intends to develop a personal coach (e-coach) that in association with the professional assistance could help anyone to choose a food for a meal, and a deep reinforcement learning system shall suggest the next meal or food composition based on the optimal policy. The training procedure will consider the nominal intake calories daily[1] (based on the Mifflin-St Jeor Equation for man [Eq. 1] and for woman [Eq. 2]) as the main goal and also, use the macronutrients[2] proportion to help this person to have a better food intake behavior.

The value obtained from Eq. 1 and Eq. 2 is the value of the basal metabolic rate (BMR) and this is the estimated number of calories a person can consume in a day to maintain their body-weight assuming they remain at rest. This value is multiplied by an activity factor (generally between 1.2 and 1.95).

[1] https://www.calculator.net/calorie-calculator.html

[2] https://www.bodybuilding.com/fun/macronutrients_calculator.htm

$$BMR = 10 \times weight(\text{kg}) + 6.25 \times height(\text{cm}) - 5 \times age(\text{years}) + 5 \quad (1)$$

$$BMR = 10 \times weight(\text{kg}) + 6.25 \times height(\text{cm}) - 5 \times age(\text{years}) - 161 \quad (2)$$

For example, for a woman with 60 kg (weight), 168 cm (height), 30-year-old and moderate activity (1.55) the BMR and Total intake calories is:

- BMR = 1,505
- Total Calories = 2,332 / day

## 3 Datasets and Inputs

This project will use a subset from "USDA National Nutrient Dataset"[3] with nutritional information (Calories, Protein, Total Fat, Carbohydrate, Fiber) for each food item used for the meal planning. Also, the Kaggle Open Food Fact dataset[4] will be used as complementary source of information to group food based on their nutritional information[5].

From USDA National Nutrient Dataset, I choosed a subset (5662 items) with these content (SR-Legacy, 2018):

- Dairy and Egg Products (291); Fats and Oils (216);
- Soups, Sauces, and Gravies (254); Sausages and Luncheon Meats (167);
- Breakfast Cereals (195); Fruits and Fruit Juices (355);
- Pork Products (336); Vegetables and Vegetable Products (814);
- Nut and Seed Products (137); Beef Products (954);
- Legumes and Legume Products (290); Baked Products (517);
- Sweets (358); Cereal Grains and Pasta (181);
- Fast Foods (312); Snacks (176); Restaurant Foods (109);

## 4 Solution Statement

A Deep Q-Learning algorithm will be used for generating the optimal policy. The state-action pair will be composed by amount of calories intake by a specific food or meal (Actions). The reward from environment will the sum of calories for a specific day, also an extension could be set the total reward for a hole week instead of only day achievement.

## 5 Benchmark Model

As benchmark, I propose a uniformly random agent and compare both using the average cumulative reward collected over the 50,000 runs for each agent.

[3] https://ndb.nal.usda.gov/ndb/search/
[4] https://www.kaggle.com/openfoodfacts/world-food-facts/home
[5] https://www.kaggle.com/lwodarzek/nutrition-table-clustering

# 6 Evaluation Metrics

- Average cumulative reward, for a high number of simulations (e.g. 100,000) for some combinations of Age, Sex, Stature (cm), Actual Weight (kg), Activity level (sedentary, low, medium, intense) which will be used for calories intake goal (Mifflin-St Jeor Equation).
- Mean Absolute Difference (MAD), for calories intake goal and cumulative reward by combination of factors.

# 7 Project Design

For this project implementation I choose to work with Python 3 language and the libraries enumerated below:

## 7.1 Libraries

- Scikit-learn
- Pandas
- Numpy
- Keras
- Tensorflow

## 7.2 Implementation Classes

The Agent and Environment will be coded as Python Classes alongside with the data from USDA National Nutrient Dataset that will be pre-processed and used for collecting and extract nutritional information for each food and accessible for both Agent and Environment as a dictionary.

The information about macronutrients (Protein, Total Fat, Carbohydrate) will be used as components for the reward functions with respective penalizations in association with the Calories (Energy) information for each food component.

```
class DQLAgent(object):
    def __init__(self, states_sz, actions_sz):
        """ Initialise the Agent """

    def _build_model(self):
        """ Build Neural Net for Deep-Q-Learning """

    def remember(self, state, action, reward, next_state, done):
        """ Implement experiece replay dataset """

    def action(self, state):
        """ Return an action from current state """

    def replay(self, batch_sz):
        """ learn using replay experiece """

    def load(self, name):
```

```python
        """ load weights """

    def save(self, name):
        """save weights """
```

The environment will be implemented as an OpenAI gym[6] environment.

```python
class SIMEnv(object):
    def __init__(self, proxy_table):
        """ Initilise the Environment with informations from proxy_table """

    def step(self, action):
        """ Take a action, calculate the reward and return these informations """

    def reset(self):
        """ Reset the Environment to the initial state """

    def seed(self, seed=None):
        """ Define the Environment seed """

    def _take_action(self, action):
        """ Select all informations from proxy table (Fat, Carbo, Protein, Energy) """

    def _get_reward(self):
        """ calculate the reward based on nutritional information """
```

## 7.3 Machine Learning Design

- **Training experience**: Experience Replay
- **Target Function**: $Q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, where $\mathcal{S}$ is the states set and $\mathcal{A}$ is the actions set; the $\mathbb{R}$ represents the value of taking a action $a \in \mathcal{A}$ beign in a state $s \in \mathcal{S}$, and following policy $\pi$.
- **Learning Function**: Deep Neural Network representation, as an function approximator.
- **Learning Algorithm**: Q-Learning, a model-free online off-policy algorithm

[6] https://gym.openai.com/