# Are Prototypical Networks good few-shot learners?

**Adi Hoftman, Yehonatan Escojido, Samuel Harroch, Dvir Shaul**
**Supervised by: Dr. Roi Weiss**

23

## 1.Project goal

Study a recently proposed deep learning method for few-shot learning in a more challenging problem than it was introduced.

## 2. Introduction

**Learning to learn**: In AI applications, agents should be able to acquire many skills and adapt to many environments.
We cannot afford to train each skill in each setting from scratch. Instead, we need our agents to **learn how to learn** new tasks faster by reusing previous experience

**Few-shot learning:** A learning problem where one needs to generalize from many previously seen solved tasks, to a new previously **unseen** task, given only a highly limited number of solved instances from the new task.

**Deep learning models** comprise of multi-layered neural networks and result in a non-linear transformation of the data. Carefully trained deep models outperform humans at many tasks.

## 3. Prototypical Networks

**Prototype rules are** based on the concept that there exists an embedding in which several points cluster around a single prototype representation for each class.

**The Prototypical networks method works as follow:** A classifier must be adapted to accommodate new classes that are not seen in training and given only a few examples of each of these classes .That it connects exactly to the concept of learning to learn since we do not practice on a lot of data but practice on understanding the data.

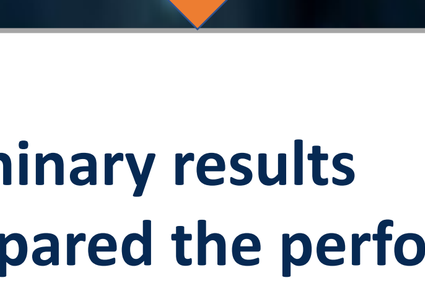
## 4. Preliminary results

**We compared the performance of the following algorithms on prediction with notMNIST as our data**

**K-NN :** the entire training dataset is stored and when a prediction is required, the k-most similar records to a new record from the training dataset are then located. From these neighbors, a summarized prediction is made.
**AK-NN :** an adaptive version of K-NN.
**Prototypical network:** trained to embed samples features in a vectorial space, in particular, at each episode (iteration), a number of samples for a subset of classes are selected and sent through the model.

## Results:

K-NN gave us 87.92 % success in prediction

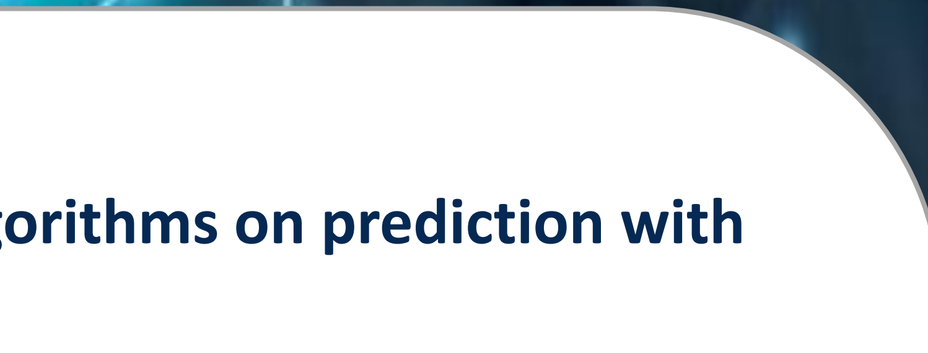



```
Max K scores: 5
Max knn Accuracy : 0.8792992950224311

knn Accuracy : 0.8792992950224311

Knn Matrix:
[[394   5   5   6   2   2   3  14   7   5]
 [ 10 399   4  24  14   6   5   0  12   2]
 [  5   5 435   2   8   3   8   0   3   0]
 [ 17  20   3 408   2   2   1   1   5   3]
 [  7   8  23   1 382  16   1   1   9   0]
 [  9   7   0   2  14 423   2   0   9  12]
 [ 16   7  40   3   8   0 398   2   3   4]
 [ 20  13   0   5   5   3   1 420   5   2]
 [  5   5   1   5   3   4   4   2 451  24]
 [ 12   2   1   3   4   3   0   1  14 406]]
```

AK-NN gave us 87% success in prediction

```
[[44  0  1  0  0  0  0  0  0  0]
 [ 3 46  0  1  0  0  0  0  0  0]
 [ 3  0 52  0  0  0  0  0  0  0]
 [ 3  2  0 41  0  0  0  0  1  0]
 [12  0  1  0 38  1  0  0  1  0]
 [ 2  0  0  0  0 39  0  0  1  1]
 [ 5  0  0  0  0  0 44  0  0  1]
 [12  0  0  0  0  0  0 37  0  0]
 [ 4  0  0  0  0  0  0  0 44  3]
 [ 3  0  0  0  0  0  0  0  2 52]]

accuracy                           0.87       500
```

Prototypical network gave us 89.25% success in prediction.




```
Test results -- Acc: 0.8925
```

Scan QRCode for full Instructions(github)

SCAN ME