Hybrid Prototypical Networks Augmented by a Non-linear Classifier

Anas El Ouardi, Maryem Rhanoui, Anissa Benlarabi and Bouchra El Asri
IMS Team, ADMIR Laboratory, Rabat IT Center, ENSIAS, Mohammed V University, Rabat, Morocco

Keywords: Text Classification, Meta-learning, Few-shot Learning.

Abstract: Text classification is one of the most prolific domains in machine learning. Present in a raw format all around us in our daily life Starting from human to human communication mainly by the social networks apps, arriving at the human-machine interaction especially with chatbots, text is a rich source of information. However, despite the remarkable performances that deep learning achieves in this field, the cost in therm of the amount of data needed to train this model still considerably high, adding to that the need of retraining this model to learn every new task. Nevertheless, a new sub-field of machine learning has emerged, named meta-learning it targets the overcoming of those limitations, widely used for image-related tasks, it can also bring solutions to tasks associated with text. Starting from this perspective we proposed a hybrid architecture based on well-known prototypical networks consisting of adapting this model to text classification and augmenting it with a non-linear classifier.

1 INTRODUCTION

When getting to learn a new task with few training examples and then using this gained knowledge to learn a different task, the human brain is the most powerful and efficient tool suitable for this purpose. For the lasts decades, artificial intelligence algorithms were given the ultimate objective of mimicking human intelligence, achieving performances in some tasks that exceed it. However, arriving at the prior cited characteristics of human intelligence these algorithms stand far from their objective since deep-learning models need huge amounts of data to learn a singular task and can not exploit this acquired knowledge to learn a new task.

In the optic to resolve this issue a sub-domain of machine learning has emerged called Meta-learning, it consists of training models on small data sets and over multiple tasks at the same time seeking to acquire the ability to generalize over new unseen tasks not over new data as in regular machine learning models(Mishra et al., 2017).

Meta-learning models are grouped under 3 major categories :

- **Model-based.** This category of models table on their internal architecture to achieve a fast convergence using a few training examples, (Santoro et al., 2016) introduced one of the most successful approaches in this category named Meta-Learning with Memory-Augmented Neural Networks consisting in augmenting the Neural Turing Machines (Graves et al., 2014) with external memory.
- **Optimization-based.** Generally composed of 2 related neural networks, the learner that takes in input data and produced the gradients of the losses and the meta-learner that takes these gradients as input producing the updated weights for the learner, playing a role similar to the role of the optimizer in classical machine learning models. The architecture proposed by (Ravi and Larochelle, 2016) follows this description where the learner is a convolutional neural network (CNN) based and the meta-learner is a long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) based neural network, however for the training of this architecture they introduced an episodic manner (that can be seen in details in 2.2). In the same category, we can also cite (Mishra et al., 2017) with their proposition that combines temporal convolutions and causal attention.
- **Metric-based.** The core idea of this last group is the use of a model capable of approximating a projection function from the input space to an embedding space (E.S) where all elements belonging to the same class should be near to each other. The proximity in the (E.S) is measured by a distance metric like euclidean distance (Snell et al., 2017; Vinyals et al., 2016).

El Ouardi, A., Rhanoui, M., Benlarabi, A. and El Asri, B.
Hybrid Prototypical Networks Augmented by a Non-linear Classifier.
DOI: 10.5220/0010498307900794
In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) - Volume 1*, pages 790-794
ISBN: 978-989-758-509-8

One of the domains that can be modeled as a meta-learning problem is text classification, been one of the major tasks in natural language processing. Many papers (Wang, 2018; Sun et al., 2020; Yamada and Shindo, 2019; Yang et al., 2019; Zaheer et al., 2021) has tackled this problem with a deep architecture that needs larges datasets to be trained one, large datasets that can not be found in the majority of fields where text classification is solicited, also these models are task-specific which means that once trained for a task they can not be used to generalize over an other task. Taking into account these elements the need for a meta-learning model for text classification can be felt.

In this paper we introduced a new simple architecture deriving from metric-based meta-learning, seeking to propose a solution for the lack of data in the text classification field. Our proposal is based on the prototypical networks (Snell et al., 2017) we adapted this model on the few-shot text classification task by adding an embedding layer aiming to transform sentences received in input to a usable format and by substituting the softmax layer applied to distance vector (between the classes centers and the query examples) after the passage of the query examples through the prototypical network by a non -linear classifier.

2 BACKGROUND

In this section, we introduce the main notions and definitions necessary to understand the framework and the context of our proposal.

2.1 Few-shot Classification

Few-shot learning (FSL) is the most known meta-Learning problem. It's a powerful paradigm dealing with tasks suffering from the lack of training examples, it involves training models on a bunch of similar tasks and testing their ability to generalize over new different tasks. Unlike deep learning classical models that require huge amounts of data to train on a single specific task and in which the knowledge gained in the training is not usable to generalize over new tasks, few-shot architectures are trained over a large number of similar tasks (text classification for example) with few examples for each task and the effectiveness of these models measured on their ability to generalize over new but similar tasks. Few-shot classification is the most important application of the FSL, consisting of applying FSL to achieve classification tasks. In recent years few-shot classification has been highly correlated with the notion of episodes. Introduced by (Vinyals et al., 2016) an episode can represent a classification task composed of the training set (support set) and the testing set (query set). The support set contains K examples from N classes sampled for every episode. We thus talk about N-way-K-Shot classification.

2.2 Task Definition

In the machine learning/deep learning field Datasets used to train the models are regularly sampled into 3 sub-sets: The train set D_{train} used to update the weights of the model during the training, the test set D_{test} disjointed from the training set used to evaluate the generalization power of the model at the end of the training and the validation set D_{val} used to select the hyper-parameters of the model before the training or to approximate the generalization power of the model during the training.

However, for few-shot classification due to the lack of data another sampling strategy that introduces the notion of metasets is used. A metaset M is composed from the two main regular sub-sets (D_{train}, D_{test}). We thus talk about new type of sets, the meta-train, meta-test and meta-validation sets (M_{train}, M_{test} and M_{val} respectively). To compose these metasets, a widely adopted strategy proposed by (Vinyals et al., 2016) is applied, it consists in sampling at every epoch a $n_{episodes}$ number of episodes from a M_j set (j can be train, test or val), such that for every episode we select a subset V of N classes from L_j set of classes available in M_j.
V will be used to compose the support set S ($S = D^j_{train}$) by selecting randomly K elements from each class present in V from the M_j in the same way the query set Q ($Q = D^j_{test}$) will be composed only this time we select t elements from every class in V.

2.3 Prototypical Networks

Prototypical networks for few-shot learning is a powerful algorithm proposed by (Snell et al., 2017). This model belongs to the metric meta-learning models and despite its simplicity, it achieves respectable performances in k-shot learning task (achieves the state of the art in zero-shot learning).

A prototypical network (Snell et al., 2017) is based on the idea that there is an embedding space where all the elements that belong to the same class represented in this space are grouped around a single prototype representative of this class. The prototype of a class c_n is calculated using the equation 1:

$$c_n = \frac{1}{|S_n|} \sum_{(x_i, y_i) \in S_n} \Psi_\theta(x_i) \quad (1)$$

where $S_n = (x_1, y_1) \dots (x_K, y_K)$ is the subset elements that belongs to the class n in S and Ψ is the embedding function.

The algorithm learns a non-linear mapping Ψ of the input to the embedding space (**E.S**) and use a convolutional neural network to estimate Ψ parameters θ.

Every query element q represented in the (**E.S**) will be affected to one class of the **N** classes available that have the highest probability calculated by the formula 2:

$$P(u_i = n \mid q_i) = \frac{\exp(-d_{bis}(\Psi_\theta(q_i), c_n)}{\sum_{n'} \exp(-d_{bis}(\Psi_\theta(q_i), c_{n'})} \quad (2)$$
$$\text{with} \quad (q_i, u_i) \in Q$$

The loss function is then computed using the equation 3 for every episode and then used to update the ϕ parameters of the embedding function Ψ_ϕ.

$$J(\theta) = -\frac{1}{T} \sum_{i}^{T} \log(P(u_i = k \mid q_i)) \quad (3)$$

3 RELATED WORKS

In the few last years, meta-learning has emerged as modeling for an artificial intelligence more human intelligence-like and a solution for lack of data problem while training deep learning models. (Vinyals et al., 2016) produced Matching Networks as a meta-learning architecture for few-shot and one-shot learning. Similar to a weighted K-nearest neighbor it aims to classify new elements based on a small support set using the cosine distance. (Snell et al., 2017) proposed the Prototypical Networks (Proto-net) which learn projection function from the input space to an embedding space where every new unseen element is grouped around a unique prototype relative to each class. This prototype represents the mean vector of the projections of the support points representative of every class. A loss function based on squared Euclidean distances was used to train this network. At the opposite of the two previous methods which uses fixed metric (cosine and euclidean distances) to discriminate unlabeled elements, the Relation Network consists on using a deep neural network to learn a non-linear distance metric which has the utility of properly classifying the query examples based on the given support examples (Sung et al., 2018).

Unlike few-shot image classification, the text few-shot classification was not widely discussed in the literature, For instance, (Gao et al., 2019) tackled the problem of noisy few-shot text classification (Relation classification) by creating hybrid network attention based on prototypical networks, composed of three components: the first one is an Instance Encoder used to encode the sentences, the second one is the Prototypical Network to compute the classes prototypes and the last one is called Hybrid Attention composed of two modules instance-level attention and feature-level attention that aims to speed-up the convergence and make the classification more reliable (Gao et al., 2019).

In the same direction, (Sun et al., 2019) proposed a modified version of the (Gao et al., 2019) architecture, especially in Hybrid Attention level making it composed of three components (Feature Level Attention, Word Level Attention, and Instance Level Multi Cross Attention) make it less sensitive to noise. These architectures integrate a lot of components that made them more sophisticated for catching text semantic however they are very complex models when arriving at decision making since.

Starting from the same philosophy we adapted the Proto-net to text classification tasks by adding an embedding layer at its beginning based on the Word2Vec (Mikolov et al., 2013), at the opposite of (Sun et al., 2019; Gao et al., 2019) who used in this layer the Glove encoding (Pennington et al., 2014) and also we augmented the Proto-net with a non-linear classifier seeking to generate the probability vector making our architecture more simple than what the two had proposed.

4 THE PROPOSED APPROACH

In this section, we introduce the three components of our model namely the **Instance encoder** that has the role of formatting the input data from the raw textual shape to the vector shape, the **Prototypical Network** which is the kernel of the model the aim of this component is to produce the distance metric vector between the classes prototypes and the queries elements, and the last part of our model **Non-linear Classifier** that generate the probability distribution over the classes.

4.1 Instance Encoder

An instance is a sentence, by definition, it's composed of a sequence of words. In there, raw format (string) words are impossible to fit into a neural network for the classification tasks purpose forexample. To remedy this issue, every word is mapped to a unique real numbers vector that catches its semantic. Let x be an

instance $x = \{w^1, w^2, ..., w^t, w^{t+1}, ..., w^T\}$ of T words, we produce an embedding vector v_i of each word instance x by applying the Word2Vec algorithm to every w^j word of this instance (Mikolov et al., 2013) such as:

$$v_i = V(\{w^1, w^2, ..., w^t, w^{t+1}, ..., w^T\}) \quad (4)$$

where V is the projection function used by the Word2Vec algorithm .

4.1.1 Prototypical Network

As seen previously in 2.3 the prototypical networks consist in generating a unique prototype specific to each class c_n by computing the average vector of all the embedding vectors v_i present in the support set belonging to this class following the equation 1.

After this step, the algorithm produces a distribution vector over the distance set between the projection of the query element q in the embedding space and the classes prototypes following the equation 2.

The training then consists in minimizing the loss function 3 by approximating the projection function to an embedding space where the queries are grouped around the prototype c_n of their respective classes and far from the other classes prototypes.

In our approach we keep the same architecture except that we skip the step of getting the probability distribution using the softmax function, keeping only the vector of distances at this level.

4.1.2 Non-linear Classifier

At this part, the model received a metric vector containing the distances between the element q and the center of all classes c_k present in the episode using the euclidean distance. The goal here been the use of this distance metric to calculate a probability distribution necessary to classify q the original proto-net uses the softmax function to do this, Contrary we propose to push furthermore this idea by adding a neural network before the softmax unit constituting the Non-linear Classifier the target of this network is to train over the distances received at every episode. Using an empirical way, the following configuration shows the most promising results. As we can see in Figure 1 our classifier is composed of an LSTM layer followed by a dropout layer (to control the overfitting), a fully connected layer, and at the end a softmax cell that generates the probability distribution.

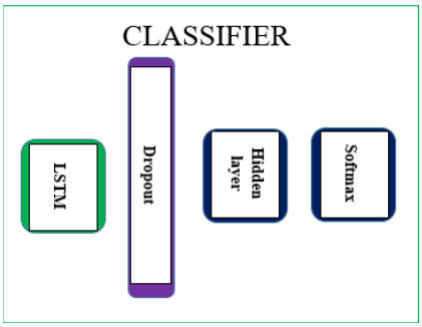

Figure 1: Non-linear Classifier.

5 CONCLUSIONS

In this paper, we proposed a prototype of a hybrid architecture based on the prototypical networks, by first adding to it an embedding layer making it compatible with text classification tasks and then augmenting it with a non-linear classifier seeking to collect more information while training necessary to generate the best probability distribution over classes. We tested our proposed model on a dataset that we composed from user tweets and got promising results. Our next step consists of testing this model over the benchmark datasets to prove its efficiency.

REFERENCES

Gao, T., Han, X., Liu, Z., and Sun, M. (2019). Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6407–6414.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *CoRR*, abs/1410.5401.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. proceedings of the international conference on learning representations (ICLR 2013).

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. (2017). A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Ravi, S. and Larochelle, H. (2016). Optimization as a model for few-shot learningg. proceedings of the international conference on learning representations (ICLR2017).

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and

Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR.

Snell, J., Swersky, K., and Zemel, R. S. (2017). Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175.

Sun, S., Sun, Q., Zhou, K., and Lv, T. (2019). Hierarchical attention prototypical networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 476–485, Hong Kong, China. Association for Computational Linguistics.

Sun, Z., Fan, C., Sun, X., Meng, Y., Wu, F., and Li, J. (2020). Neural semi-supervised learning for text classification under large-scale pretraining.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., and Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638.

Wang, B. (2018). Disconnected recurrent neural networks for text categorization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2311–2320.

Yamada, I. and Shindo, H. (2019). Neural attentive bag-of-entities model for text classification. *CoRR*, abs/1909.01259.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. (2021). Big bird: Transformers for longer sequences.