Recurrent Neural Networks for Adaptive Feature Acquisition

Thierry Artières, Gabriella Contardo, Ludovic Denoyer

Recurrent Neural Networks for Adaptive Feature Acquisition

Gabriella Contardo[1], Ludovic Denoyer[1], and Thierry Artières[2]

[1] Sorbonne Universités,UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris.
[2] Aix Marseille Univ, CNRS, Centrale Marseille, LIF, Marseille, France.

Abstract. We propose to tackle the cost-sensitive learning problem, where each feature is associated to a particular acquisition cost. We propose a new model with the following key properties: (i) it acquires features in an adaptive way, (ii) features can be acquired *per block* (several at a time) so that this model can deal with high dimensional data, and (iii) it relies on representation-learning ideas. The effectiveness of this approach is demonstrated on several experiments considering a variety of datasets and with different cost settings.

1 Introduction

The development of attention models [13,1,14] is a recent trend in the neural network (NN) community. It usually consists in adding an attention module to classical NN architectures which goal is to select relevant information to use for predicting instead of using the whole input. These models have been mainly developed for particular types of data i.e images and text and are specific to the nature of the inputs. More generally, the objective of selecting relevant information is not new and different models have been proposed in the Machine Learning domain during the last decades e.g. L1 regularization (e.g [3]) or dimensionality reduction techniques [10]. These works are mainly motivated by the need to not only select relevant input information – as it is the case for attention models – but also to limit the inference cost in applications where the acquisition or computation of input features is expensive. Applications like medical diagnosis or personalized predictive tasks are intuitive examples of such setting, where some input features can be very expensive (e.g fMRI exams). One can also think of many today applications such as spam detection ([17]), web-search ([24,6]), where one wants to answer huge numbers of prediction per second, per minute or per day. In that cases, limiting the number of input features used for prediction is a key factor for an algorithm, in order to constraint the "cost" of the information used, while keeping robust prediction ability. An optimal strategy to limit this cost should rely on an adaptive feature acquisition process (as in attention models), i.e selecting features according to what has been currently observed of the input, since it is quite likely that not all inputs require the knowledge of the same subset of features to perform an accurate prediction.

We propose a new sequential model based on a recurrent neural network architecture to tackle this problem of cost-sensitive features selection. At each

time-step, the model chooses which features to acquire and builds a representation of the partially observed input based on all the acquired information. This learned representation is then used to both drive the future acquisition steps, but also to compute a final prediction at the end of the acquisition process. Our algorithm is thus an adaptive one which is able to select different subsets of features depending on the input and is learned based on the objective to find a good trade-off between the average acquisition cost and the quality of the prediction. At the opposite to recent NN-based techniques, our model is not specific to a particular nature of the data, and the attention part of the model is guided by the cost of the different features. Moreover, our algorithm is able to acquire multiple features at each timestep making it scalable for dealing with high dimensional data. These key aspects – adaptiveness, ability to handle different cost for different features, scalability, and representation learning based approach – are, to the best of our knowledge, novel in regard of the state of the art (see Section 4).

The paper is organized as follows: Section 2 details the cost-sensitive acquisition problem and details our RNN model and experimental results are provided in Section 3. Section 4 situates our work with respect to state of the art.

2 Cost-Sensitive Recurrent Neural Network

We consider the generic problem of computing a prediction $y \in \mathbb{R}^Y$ based on an input $x \in \mathbb{R}^n$ where n is the dimensionality of the input space, y is an output vector, and Y is the dimension of the output space. x_i (resp y_i) denotes the i-th features of x (resp. y). We particularly focus on the classification task where Y is the number of possible categories, and $y_i = 1$ if the input belongs to category i and $y_i = -1$ elsewhere.

2.1 Recurrent ADaptive AcquisitIon Network (RADIN[3])

The generic principle of adaptive feature acquisition may be resumed as follows. A model starts by acquiring a first subset of features from an input x. Then, new features are iteratively acquired at each timestep based on what has already been observed, we note T the number of steps made by the model. The final prediction is then performed based on the set of acquired features. Many models can be cast in this formalism. Non-adaptive feature selection approach stands for one step models ($T = 1$), while a decision tree may be thought as starting in the root node and acquiring a new feature one at a time that depends on the node in the tree.

We propose in this work to instantiate this general framework with a Recurrent Neural Network architecture (see Figure 1). At each timestep, the acquired information enrich a latent representation of the input, and further decisions (acquisition of new features and final prediction) are made based on this representation. The internal state of the RNN (i.e. a continuous vector in $z \subset R^p$, p

[3] In french, "radin" means "skinflint"

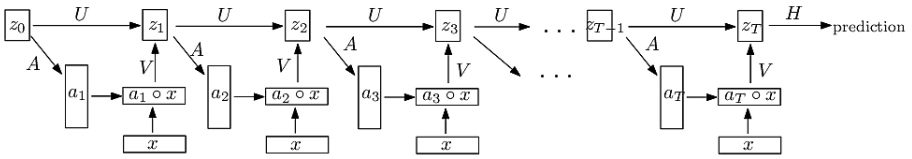

Fig. 1: Architecture of our recurrent acquisition network.

being the dimensionality of the latent space) is used to encode the information gathered on an input sample x through a subset of observed features. It is initialized as the null vector $z_0 = 0^p$. It is enriched all along the acquisition process, yielding a series of representations, z_1, z_2... up to a final iteration T and a final representation z_T. The use of multiple steps enables data dependent feature acquisition. The final representation of x, z_T, is used to perform prediction. It is worth noting that z_T is built from a partial view of x, i.e. only a subset of its features have been observed when performing prediction.

We discuss now the RNN architecture and how z_t's are updated. The underlying mechanism involve both an *attention layer* in charge of choosing which features to acquire, and an *aggregation layer* in charge of aggregating the newly acquired information to the previously collected features.

Attention Layer: While in classical RNN, the input at time t is usually a predetermined piece of the input (an element of an input sequence for example), in our case, this input is chosen by the model as a function of the previous state z_{t-1} in the following way: A specific *attention layer* computes a vector $a_t = f(A \times z_{t-1}) \in [0,1]^n$ whose component i denoted $a_{t,i}$ stands for the usefulness of feature i of the input denoted x_i. a_t is an attention vector that aims at selecting the features to acquire i.e the features i such that $a_{t,i} > 0$. This vector is computed based on the previous representation z_{t-1} and different inputs will thus produce different values of the attention layer resulting in an adaptive acquisition model. f is typically a non-linear activation function and $A \in \mathbb{R}^{n \times p}$ corresponds to the parameters of the attention layer. In order to compute the input of the hidden layer, the attention vector is then "mixed" with the original input x by using the Hadamard product[4], the attention layer acting as a filter on the features of x. This input is denoted $x\,[a_t] = a_t \circ x$ in the following. Note that in the particular case where a_t would be a binary vector, this stands for a copy of x where features that should not be acquired are set to 0. This vector $x\,[a_t]$ is an additional input that is used to update the internal state, i.e. to compute z_t.

Aggregation layer: Once newly features have been acquired, the internal state z_t is updated according to $z_t = f(U \times z_{t-1} + V \times x\,[a_t])$ (with U and V

[4] Note that the Hadamard product is used during training since the training inputs are fully known. During inference on new inputs, the value of the Hadamard product is directly computed by only acquiring the chosen features.

Corpus Name	Nb.Feat.	Nb.Cat.	Feature used (%)	Model			
				SVM L_1	DT	GreedyMiser	RADIN
Cardio	21	10	90 %	0.683	0.775	**0.827**	0.824
			75 %	0.580	0.775	**0.825**	**0.825**
			50 %	0.496	0.775	0.751	**0.837**
			25 %	0.338	0.771	0.508	**0.775**
			10 %	0.259	0.643	0.325	**0.662**
Statlog	60	3	90 %	0.775	0.823	0.851	**0.859**
			75 %	0.741	0.823	0.846	**0.858**
			50 %	0.703	0.823	0.831	**0.858**
			25 %	0.630	0.823	0.765	**0.852**
			10 %	0.587	0.821	0.605	**0.833**
MNIST	780	10	90 %	0.897	0.808	0.920	**0.950**
			75 %	0.897	0.808	0.920	**0.948**
			50 %	0.882	0.808	0.903	**0.926**
			25 %	0.704	0.808	0.846	**0.920**
			10 %	0.577	0.808	0.776	**0.859**
gisette	5000	2	25 %	**0.970**	0.919	0.884	0.957
			10 %	**0.968**	0.919	0.884	0.957
			5 %	**0.963**	0.919	0.867	0.957
			1 %	0.910	0.919	0.785	**0.947**
r8	6224	8	25 %	**0.969**	0.901	0.948	0.962
			10 %	**0.968**	0.901	0.947	0.961
			5 %	0.951	0.901	0.945	**0.961**
			1 %	0.913	0.901	0.939	**0.959**
webkb	5388	4	25 %	0.891	0.793	0.861	**0.962**
			10 %	0.887	0.793	0.864	**0.961**
			5 %	0.859	0.793	0.857	**0.865**
			1 %	0.717	0.793	0.828	**0.831**

Table 1: Results of the different models w.r.t percentage of features used on different datasets.

two weight matrices of size $p \times p$ and $p \times n$) as in classical RNN cells [5]. The internal state layer z_t is thus an aggregation of the information gathered from all previous acquisition steps up to step t.

Decision Layer: The final representation z_T, which is obtained after the T-acquisition step, is used to perform classification $o(x) = g(H \times z_T) \in R^Y$ with g a non linear function and H a weight matrix of size $Y \times p$, z_T being the representation of the input x at the end of the acquisition process.

Noting c_i the acquisition cost for feature i, $c_i \geq 0$ and $c \in \mathbb{R}^n$ the vector of all feature costs, the quantity $\sum_{t=1}^{T} a_t^\intercal.c$ stands for the actual acquisition cost provided that $a_{t,i}$ are actually binary values and that a feature cannot be acquired twice. In order to train the RNN to learn to acquire efficiently information from the inputs before classifying it we propose to optimize the weight parameters A, U, V, H to minimize the following empirical loss on a set of N training samples $(x^k, y^k)_{k=1..N}$:

$$\mathcal{J}^{emp}(A, U, V, H) = \sum_{k=1...N} \left[\Delta(o(x^k), y^k) + \lambda \sum_{t=1}^{T} a_t^\intercal.c \right] \quad (1)$$

where the first term of the loss is a data fit term that measures how well prediction is performed on training samples and the second term is related to the

[5] We also tested Gated Recurrent Unit ([8]).

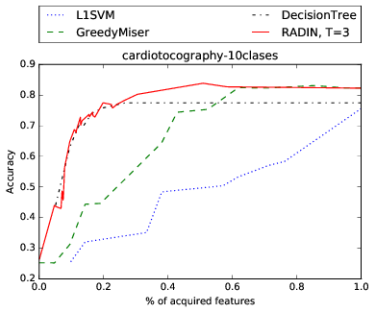

Fig. 2: Accuracy/Cost on *cardio*.

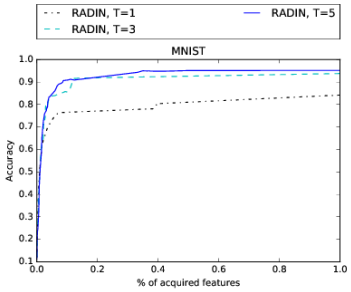

Fig. 3: RADIN with different T values on *MNIST*.

constraint on the feature acquisition *budget*. In practice however, dealing with binary attention vectors a_t leads to a difficult optimization problem so that we use for our model continuous values $a_{t,i} \in [0, 1]$ with a similar meaning i.e x_i is acquired only if $a_{t,i} > 0$. In that case, the regularization term is an approximation of the *budget term* that acts as a penalty term that drives $a_{t,i}$ towards 0. This continuous relaxation is close to what it is usually done when using L_1-regularized models instead of L_0 ones.

It is worth mentioning here that the architecture we present allows feature acquisition to be performed **per block**, i.e. many features at a time (in one step), as a_t can have several non-null values. This is an interesting, and quite novel property with regard to state of the art methods for (cost-sensitive) problems, as it allows this model to scale well to data with a large number of features, reaching high accuracy while keeping a reasonable computational complexity of the process.

3 Experiments

This section provides results of various experiments on feature-acquisition problems with different cost-settings. We study the ability of our approach on several mono-label classification datasets [6]. Let us first describe our **experimental protocol** for validation (as our goal is both to optimize the accuracy as well as the acquisition cost, usual cross-validation protocol cannot be conducted here). Each dataset is split in training, validation and testing sets [7]. We then learn several models with various hyper-parameters settings on the *training set*. Each learned model yields a two dimension point (accuracy,cost) on the *validation set*. The Pareto Front of this set of points is then computed to select the best models. At last, the selected models are evaluated on the *test set* on which results are reported. We used the following specifications for our model: a linear function for prediction o , GRU or RNN cells for the aggregation layer, and a hard logistic activation function for the attention layer. Δ is a mean-square error. The code used

[6] Note that our approach also handles other problems such as multi-label classification, regression or ranking as long as the loss function Δ is differentiable.

[7] One third of the examples for each set, except for MNIST, where the split corresponds to 15%,5%,80% of the data

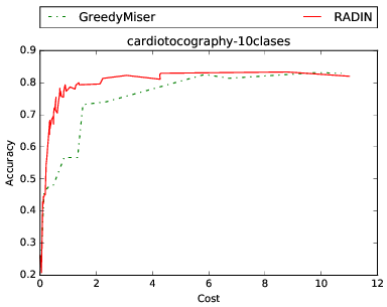

Fig. 4: Cost-sensitive setting on *cardio* dataset

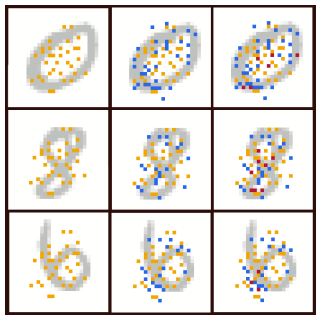

Fig. 5: Illustration of the adaptive behavior of RADIN on three different MNIST inputs

to conduct these experiments is available at `http://github.com/ludc/radin`. We compare our model with three different approaches : (i) a L_1-regularized linear **SVM**, (ii) a **Decision Tree**, (iii) a cost-sensitive method that constraints locally and globally the cost of a set of weak classifiers -decision trees- **GreedyMiser** ([22]). Note that the first two methods can't handle *cost-sensitive* problems. Due to a lack of space, we do not present here all the results obtained on many different datasets, and just focus on the more representative performance.

Let us first focus on experiments with **uniform cost**, i.e $\forall i, c_i = 1$. In this case, the acquisition cost is directly the number of features gathered. The cost is thus expressed as the percentage of features acquired w.r.t the total number of available features. Figure 2 illustrates the overall accuracy-cost curves for the dataset *cardio*. One can see for example that the GreedyMiser approach yields an accuracy of about 68 % for a cost of 0.4, i.e acquiring 40 % of the features, while our model RADIN obtains approximately 82 % of accuracy for the same amount of acquired features. The results in Table 1 show the ability of our method to give competitive or better results on all datasets, including larger scale datasets, particularly in this case when the percentage of acquired features drops substantially.

The Figure 3, which plots the accuracy/cost curves obtained with different number of acquisition steps T illustrates the adaptive behavior of our model and its ability to choose relevant features depending on the input. Moreover, one can see on Figure 5 which features are acquired considering three particular inputs of the MNIST dataset. Each color corresponds to a particular acquisition step. If the features acquired at time $T = 1$ are the same, RADIN exhibits a different behavior for the following steps depending on the acquired information.

At last, we have considered a different cost-sensitive setting and show the performance obtained by RADIN and GreedyMiser on an artificial cost-sensitive dataset constructed from the *cardio* dataset, where the cost of feature i is defined as $c_i = \frac{i}{n}$. One can observe (Figure 4) that our model yields better accuracy results than Greedy Miser for all the cost range, which indicates its ability to not only acquire the relevant features but also to integrate in the process their different costs c_i.

4 Related Work

Many methods have been proposed under a *static* features selection framework, i.e with only one step of acquisition. A good overview of existing methods is provided in [10] which describe different approaches like *wrapper* methods [12] or *Embedded* methods [3,19] with l_1 and l_0-norm. Block feature selection has also been proposed but feature blocks have to be known beforehand [23]. Note that these approaches generally cannot handle non-uniform costs. Another family of algorithm proposes to tackle the features selection problem by estimating the information gain of the features [4] For example, [5] presents two greedy strategies to learn a test-cost sensitive naive Bayes classifier, while [18] propose to use reinforcement-learning to learn a value-function of the information gain. In the adaptive features selection literature, decision trees are naturally good candidates and they are used for example in [22,20] as several weak constraint classifiers. Another type of approach relies on learning *cascade* of classifiers, as in [16] and [7], the classifier being used depending on the input. More recently, [21] presented a method to learn a tree of classifiers which can be extended to cascade architecture inducing the possibility of *early-stopping*, which is an interesting aspect of adaptive prediction behavior. Feature acquisition can also be seen as a sequential decision process, and it has been studied under the MDP and Reinforcement Learning framework, as in [11], which models the problem as a partially observable MDP, or in [2,15,9] using classical RL algorithms. Here, these models usually suffer when the number of features is too large. At last, new deep learning models have recently emerged [13,1,14] and are closely related to our work. They consist in adding an attention mechanism to classical architecture. They have been mainly developed for images and text with the goal to increase the quality of the model. They thus don't consider a particular budget or cost-sensitive setting.

Regarding these various methods, our work differs on several aspects. It is, to the best of our knowledge, the first approach that tackles the adaptive cost-sensitive acquisition problem in a generic way with a RNN-like architecture.

5 Conclusion

We presented a recurrent neural network architecture to tackle the problem of adaptive cost-sensitive acquisition. Our approach can acquire the features per block and can be learned using efficient gradient descent algorithm. We showed that our model performs well on different problem settings and is able to acquire information resulting in a good cost/accuracy trade-off in an adaptive way.

Acknowledgments : This article has been supported within the Labex SMART supported by French state funds managed by the ANR within the Investissements d'Avenir programme under reference ANR-11-LABX-65. Part of this work has benefited from a grant from program DGA-RAPID, project LuxidX.

References

1. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755 (2014)
2. Benbouzid, D., Busa-Fekete, R., Kégl, B.: Fast classification using sparse decision dags. ICML (2012)
3. Bi, J., Bennett, K., Embrechts, M., Breneman, C., Song, M.: Dimensionality reduction via sparse support vector machines. JMLR 3, 1229–1243 (2003)
4. Bilgic, M., Getoor, L.: Voila: Efficient feature-value acquisition for classification. In: Proceedings of the national conference on artificial intelligence (2007)
5. Chai, X., Deng, L., Yang, Q., Ling, C.X.: Test-cost sensitive naive bayes classification. In: Data Mining,ICDM'04 (2004)
6. Chapelle, O., Shivaswamy, P., Vadrevu, S., Weinberger, K., Zhang, Y., Tseng, B.: Boosted multi-task learning. Machine learning 85(1-2), 149–173 (2011)
7. Chen, M., Weinberger, K.Q., Chapelle, O., Kedem, D., Xu, Z.: Classifier cascade for minimizing feature evaluation cost. In: AISTATS. pp. 218–226 (2012)
8. Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
9. Dulac-Arnold, G., Denoyer, L., Preux, P., Gallinari, P.: Sequential approaches for learning datum-wise sparse representations. Machine learning (2012)
10. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. JMLR (2003)
11. Ji, S., Carin, L.: Cost-sensitive feature acquisition and classification. Pattern Recognition 40(5), 1474–1485 (2007)
12. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial intelligence 97(1), 273–324 (1997)
13. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: NIPS (2014)
14. Sermanet, P., Frome, A., Real, E.: Attention for fine-grained categorization. arXiv preprint arXiv:1412.7054 (2014)
15. Trapeznikov, K., Saligrama, V.: Supervised sequential classification under budget constraints. In: AISTATS (2013)
16. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision 4, 51–52 (2001)
17. Weinberger, K., Dasgupta, A., Langford, J., Smola, A., Attenberg, J.: Feature hashing for large scale multitask learning. In: ICML. ACM (2009)
18. Weiss, D.J., Taskar, B.: Learning adaptive value of information for structured prediction. In: NIPS (2013)
19. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V.: Feature selection for svms. In: NIPS (2000)
20. Xu, Z., Huang, G., Weinberger, K.Q., Zheng, A.X.: Gradient boosted feature selection. In: ACM SIGKDD (2014)
21. Xu, Z., Kusner, M.J., Weinberger, K.Q., Chen, M., Chapelle, O.: Classifier cascades and trees for minimizing feature evaluation cost. JMLR (2014)
22. Xu, Z., Weinberger, K., Chapelle, O.: The greedy miser: Learning under test-time budgets. arXiv preprint arXiv:1206.6451 (2012)
23. Yuan, M., Lin, Y.: Efficient empirical Bayes variable selection and estimation in linear models. Journal of the American Statistical Association (2005)
24. Zheng, Z., Zha, H., Zhang, T., Chapelle, O., Chen, K., Sun, G.: A general boosting method and its application to learning ranking functions for web search. In: NIPS (2008)