Cognitive Consistency Routing Algorithm of Capsule-network

1st Huayu Li

Northern Arizona University
Flagstaff, United State
Northern Arizona University
hl459@nau.edu

Abstract—**Artificial Neural Networks (ANNs) are computational models inspired by the central nervous system (especially the brain) of animals and are used to estimate or generate unknown approximation functions that rely on a large amount of inputs. The Capsule Neural Network [1] is a novel structure of Convolutional Neural Networks(CNN) which simulates the visual processing system of human brain. In this paper, we introduce a psychological theory which is called Cognitive Consistency to optimize the routing algorithm of Capsnet to make it more close to the working pattern of human brain. Our experiments show that progress had been made compared with the baseline.**

Index Terms—**Capnet, Cognitive Consistency**

I. Introduction

Convolutional neural networks(CNN) contribute to a series of breakthroughs for image classification task. There are plenty of structures of CNN proven to make an outstanding performance in classification tasks in various domains. Capsnet is a novel variant of CNN proposed by [1]. Capsnet uses the outputs of a group of neurons which is called capsule [2] to represent different properties of the same entity. The mechanism [1] of Capsnet is to ensure that the output of the capsule gets sent to an appropriate parent in the layer above. Initially, the output is routed to all possible parents after scaled down by coupling coefficients that sum to 1. A "prediction vector" which is the product of its own output and a weight matrix for each possible parent is computed by the capsule. Top-down feedback increases the coupling coefficient for that parent and decreases coupling coefficient for other parents, determined by the scalar product of this prediction vector and the output of a possible parent. The parent further increases the scalar product of the capsule's prediction with the parent's output as a result of increasing the contribution made by the capsule. In [1], the "agreement" is mentioned as simply the scalar product $a_j = v_j \cdot \hat{u}_{j|i}$ which is treated as if it was a log likelihood and is added to the initial logit, b_{ij} before computing the new values for all the coupling coefficients linking capsule i to higher level capsules.

In this paper, we were inspired by psychological theories and applied them to develop a new method of routing algorithm. According to [3], people have a drive to generate consistent cognition and behavior on objects. When the cognition is dissonant, people will feel uncomfortable, and then try to reduce it, reducing a mechanism of dissonance by selectively seeking support information or avoiding inconsistent information.

We simply regard the prediction vectors made by each capsule as the "cognition" it keeps, and our work is to explore an algorithm to unify the cognition between each capsule, which we call the Cognitive Consistency Routing Algorithm. We propose to treat the differences between the input "prediction vectors" from the lower layer and the "prediction vectors" made by the current layer as dissonance coefficients to ensure the coupling between higher capsules and lower layer increases much sooner than if the differences between their predictions are smaller. The motivation behind the Cognitive Consistency Routing Algorithm is to ensure each capsule layer makes the prediction of the target is as consistent as possible. In other words, the capsule layers of the whole network should ultimately tend to a consistent cognition.

II. Cognitive Consistency Routing Algorithm

There are many possible ways to implement the general idea of Cognitive Consistency. The aim of this paper is not to find a method that achieves the state-of-art performance but simply to test and verify the practicability of the Cognitive Consistency Routing Algorithm.

Firstly, we want to get the initial values of the prior probabilities from the lower capsules and let the capsules discriminatively coupled to the lower layers. We therefore use a simple "clip" function for the prior probabilities initially to ensure that the capsules above accept the prediction vectors $\hat{\mathbf{u}}_{j|i}$ from the capsules below and restrict them within a range.

$$b_{ij} = \begin{cases} a_{min}, & \text{if } \hat{\mathbf{u}}_{j|i} < a_{min} \\ \hat{u}_{j|i}, & \text{if } a_{min} < \hat{\mathbf{u}}_{j|i} < a_{max} \\ a_{max}, & \text{if } \hat{\mathbf{u}}_{j|i} > a_{min}, \end{cases} \quad (1)$$

where the $\hat{\mathbf{u}}_{j|i}$ are the prediction vectors from the lower capsules and (a_{min}, a_{max}) is the expected range of b_{ij}. The motivation behind "clip" function is to avoid completely inactivating the prediction vectors. We expect the "bad guys" which are decided by the lower capsule continue to have the right to make predictions in current layer thus

the whole network maintain a consistent cognition without partial dissonance.

The coupling coefficients c_{ij} between capsule i and all the capsules in the layer above sum to 1 are determined by the distribution of the prior probabilities by the "softmax" function.

$$c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})} \qquad (2)$$

Our approach to get the total input to a capsule s_j is as the same as Sabour S, et al.[2017].

$$\mathbf{s}_j = \sum_j c_{ij}\hat{\mathbf{u}}_{j|i}, \hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i \qquad (3)$$

where $\mathbf{u}_i$ is the output of capsule in the layer below and $\mathbf{W}_{ij}$ is the weight matrix.

And the vector output of capsule j is the "squashing" results of $\mathbf{s}_j$.

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2} \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|} \qquad (4)$$

which ensure that the vectors are in the range of zero to one.

The prior probabilities are supposed to be iterated by adding the scalar product $\|\mathbf{v}_j\| \cdot \|\hat{\mathbf{u}}_{j|i}\| \cdot a_{ij}$.

$$b_{ij} = b_{ij} + |\mathbf{v}_j| \cdot |\hat{\mathbf{u}}_{j|i}| \cdot a_{ij}, a_{ij} = cos((|\mathbf{v}_j| - |\hat{\mathbf{u}}_{j|i}|)^2) \quad (5)$$

where a_{ij} are the consistency ratios that decreases with the increment of difference between the input prediction vectors from the lower layer and the prediction vectors made by the current layer. The curve of a_{ij} is shown as Fig. 1.
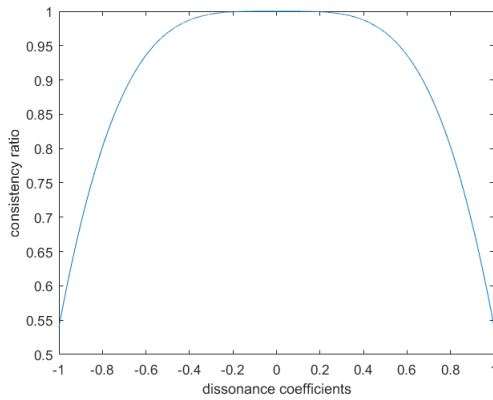

Fig. 1. The curve of consistency ratios shows that the bigger difference between the predictions made by the current layer and the lower layer lead to the smaller consistency ratios which means each layer tends to a consistent cognition to avoid cognitive dissonance.

The overall flow of our algorithm is shown in Algorithm 1.

Algorithm 1 Cognitive Consistency Routing Algorithm of Capsule-network

1: For all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow clip(\hat{\mathbf{u}}_{j|i}, a_{min}, a_{max})$.
2: **for** each r iterations **do**
3:     for all capsule i in layer l: $c_{ij} \leftarrow \frac{exp(b_{ij})}{\sum_k exp(b_{ik})}$;
4:     for all capsule j in layer $l+1$: $\mathbf{s}_j \leftarrow \sum_j c_{ij}\hat{\mathbf{u}}_{j|i}$;
5:     for all capsule j in layer $l+1$: $\mathbf{v}_j \leftarrow squash(\mathbf{s}_j)$;
6:     for all capsule i in layer l and all capsule j in layer $l+1$: $b_{ij} \leftarrow b_{ij} + \|\mathbf{v}_j\| \cdot \|\hat{\mathbf{u}}_{j|i}\| \cdot cos((|\mathbf{v}_j| - |\hat{\mathbf{u}}_{j|i}|)^2)$;
7: **return** $\mathbf{v}_j$;

III. EXPERIMENTAL RESULTS

A. Our Algorithm on MNIST

We tested our algorithm on MNIST to verify whether our algorithm works or not. Our model has the same architecture as [1] showed in Fig. 2 and was set as 3 times routing, but with batch normalization(BN) [4] after each layer. We tested two algorithms on this model and compared their performance. We observed that a Capsnet with Cognitive Consistency Routing achieves state-of-the-art performance on MNIST which equals to the baseline.
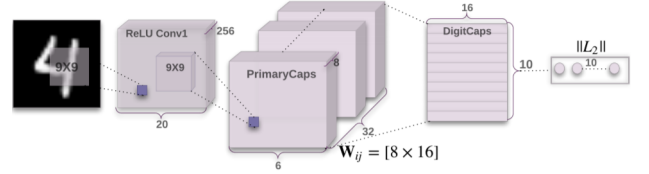

Fig. 2. The structure of the original Capsnet.

B. Our Algorithm on Fashion-MNIST

Via experiments on other datasets, we can simply observe that our algorithm get better and higher results than the baseline in more complex data. We use the same model as on MNIST. We evaluated our method and the original algorithm on the Fashion-MNIST dataset [5] which consists of 28x28 grayscale images of 70,000 fashion products from 10 categories, with 7,000 images per category. The models were trained on the 48,000 training images and evaluated on the 12,000 validation images. We also obtained a final result on the 10,000 test images. We tested using 10 models and got the final result by model averaging. We can see from table 1 that our routing approach achieved better performance than the baseline.

Iterations	1	3	5	10	15
Sabour S, et al.[2017]	0.7632	0.7993	0.8227	0.8514	0.8644
Our Algorithm	0.7832	0.8163	0.8266	0.8655	0.8803

TABLE I
OUR ALGORITHM ON FASHION-MNIST, THE HIGHER THE BETTER AS THE ORIGINAL ALGORITHM

IV. Conclusion

We show that the Cognitive Consistency from psychological theories can improve the capsule neural network and demonstrate that introducing the theories from other domains, besides only statistics or computer science is an effective way to perfect the theoretical foundation of deep learning and Artificial Neural Networks. Future work includes improving our algorithm to achieve the state-of-art performance and trying to introduce more theories from psychologic and cognitive science into Artificial Neural Networks.

References

[1] Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules[C] Advances in Neural Information Processing Systems. 2017: 3856-3866.

[2] Hinton G E, Krizhevsky A, Wang S D. Transforming auto-encoders[C] International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 2011: 44-51.

[3] Festinger L. A theory of cognitive dissonance[M]. Stanford university press, 1962.

[4] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.

[5] Xiao H, Rasul K, Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms[J]. arXiv preprint arXiv:1708.07747, 2017.