TimeCaps: Learning from Time Series Data with Capsule Networks

Supplementary Material

Hirunima Jayasekara[1], Vinoj Jayasundara[1], Mohamed Athif[2], Jathushan Rajasegaran[1], Sandaru Jayasekara[1], Suranga Seneviratne[3], Ranga Rodrigo[1]
[1] *University of Moratuwa, Sri Lanka*
[2] *Boston University, MA, USA*
[3] *School of Computer Science, University of Sydney*
{nhirunima, vinojjayasundara}@gmail.com, athif@bu.edu,{brjathu, sandaruamashan}@gmail.com,
suranga.seneviratne@sydney.edu.au, ranga@uom.lk

I. Capsule Networks and Dynamic Routing Algorithm

A capsule is a group of neurons whose activity vectors represent not only the probability of existence of a certain entity, but a set of instantiation parameters [1]. The instantiation parameters can include pose (position, size, orientation), deformation, velocity, albedo, hue, texture and etc. [2]. Hence, a capsule can be said to have the ability to learn a set of properties of a particular entity, rather than only learning to detect the presence of that entity, similar to CNN.

An important notion in utilizing part-whole relationship for object recognition is that, even though the instantiation parameters for part entities and whole entities can vary for different inputs, the spatial relationship between a part entity and its whole entity remains static. Hence, part-whole relationship is viewpoint invariant [1].

Consider that the capsules in layer l learn part entities whereas the capsules in layer $l+1$ learns the whole entities, which are connected to the part entities in layer l. Each output vector u_i of layer l, represents the instantiation parameters of the part entity. Capsules in layer $l+1$ build a whole entity capsule v_j from $u_i, i \in [N_l]$, where N_l, N_{l+1} are the number of capsules in layer $l, l+1$ respectively.

In the perspective of the capsule j, it can receive multiple predictions from the lower layer. Similarly, in the perspective of Capsule i, it has a choice of sending its output to multiple capsules in the higher layer. The fact that the output of a capsule outputs a more meaningful activity vector, rather than merely an activation value, allows the output to be dynamically routed to high level capsules which agree more with it. Such routing mechanisms are termed as *Routing-by-agreement*.

A. Dynamic Routing Algorithm

Part-whole spatial relationship between the capsule i in the layer l and the capsule j in the layer $l+1$ is learned by the transformation matrix W_{ij} during back propagation. Hence, from equation[1], we can obtain a prediction for the existence and pose of the part entity in layer $l+1$ from $\hat{u}_{j|i}$.

$$\hat{u}_{j|i} = W_{ij} \times u_i \tag{1}$$

$$s_j = \sum_i c_{ij}\hat{u}_{ij} \tag{2}$$

s_j can be interpreted as the weighted sum of predictions from the capsules in layer l. Hence, the output of the capsule j, v_j can be calculated with the aid of s_j, according to [3]. Squash function is used to keep the capsule vector within unit length while preserving orientation, so it"s length will represent the probability of the existence of an entity.

$$v_j = \frac{||s_j||^2}{1+||s_j||^2}\frac{s_j}{||s_j||} \tag{3}$$

The subsequent task is to route the outputs of the capsules in layer l as the inputs of capsules in layer $l{+}1$, which requires c_{ij} to be updated at each routing.

$$c_{ij} = \frac{exp(b_{ij})}{\sum_j exp(b_{ik})} \tag{4}$$

$$b_{ij} = b_{ij} + \hat{u}_{ij} \cdot v_j \tag{5}$$

b_{ij} is initialized to zero to provide equal value to c_{ij} in first routing. As the procedure progresses, c_{ij} will be eventually updated with $a_{ij} = \hat{u}_{ij} \cdot v_j$. Here, a_{ij} corresponds to the agreement between v_j and $\hat{u}_{ij}$, hence routing by agreement.

References

[1] Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: ICANN 2011, Berlin, Heidelberg (2011) 44–51
[2] Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: NIPS 2017. (2017) 3856–3866