An Exploration of Capsule
ADLxMLDS Final Project

b03902036 **劉彥廷**, b03902072 **江廷睿**

Hippocampus

Abstract

Recently Sabour et al. propose a new kind of neuron network layer in the paper "Dynamic Routing Between Capsules" [2]. Before long, another paper "Matrix Capsule with EM Routing" [1] was under the open review of ICLR 2018. First, we would like to reproduce the result of those capsule network. And see if it has different property compared with general CNN when visualization. For some designs of those capsule networks, we also want to see its effect when applied on general CNN. In conclusion, we did

- (Attempt to) implement capsule with EM routing.
- Visualize patterns learned by capsules.
- Experiment on matrix CNN.
- Experiment on the robustness against adversarial attack.

Background

Capsule Unit

	General Neuron	Capsule Neuron
Each neuron has	Activation $a_i \in \mathbb{R}$	Pose $p_i \in \mathbb{R}^n$or$\mathbb{R}^{n\times n}$ Activation $a_i \in \mathbb{R}$
Neuron j of upper layer	$\sum_i w_{ij} a_i$	$\frac{\sum_i r_i a_i W_{ij} p_i}{\sum_i r_i a_i}$
Weight learning	w_{ij}: discriminately	W_{ij}: discriminately r_i: routing algorithm

Table 1: Comparison between capsule network and general neural network.

Routing Algorithm

A routing algorithm decide how lower level capsules are linked to upper level capsules dynamically according to their agreement. Generally it is an iterative procedure. We can take the EM-routing algorithm in [1] for example.

procedure EM ROUTING($\boldsymbol{a}, V$)
  $\forall i \in \Omega_L, j \in \Omega_{L+1}$: $R_{ij} \leftarrow 1/|\Omega_{L+1}|$
  for t iterations **do**
    $\forall j \in \Omega_{L+1}$: M-STEP($\boldsymbol{a}, R, V, j$)
    $\forall i \in \Omega_L$: E-STEP($\mu, \sigma, \boldsymbol{a}, V, i$)
  return $\boldsymbol{a}, M$

procedure M-STEP($\boldsymbol{a}, R, V, j$) ▷ for one higher-level capsule
  $\forall i \in \Omega_L$: $R_{ij} \leftarrow R_{ij} * \boldsymbol{a}_i$
  $\forall h$: $\mu_j^h \leftarrow \frac{\sum_i R_{ij} V_{ij}^h}{\sum_i R_{ij}}$
  $\forall h$: $(\sigma_j^h)^2 \leftarrow \frac{\sum_i R_{ij} (V_{ij}^h - \mu_j^h)^2}{\sum_i R_{ij}}$
  $cost^h \leftarrow (\beta_v + log(\sigma_j^h)) \sum_i R_{ij}$
  $a_j \leftarrow sigmoid(\lambda(\beta_a - \sum_h cost^h))$

procedure E-STEP($\mu, \sigma, \boldsymbol{a}, V, i$) ▷ for one lower-level capsule
  $\forall j \in \Omega_{L+1}$: $\boldsymbol{p}_j \leftarrow \frac{1}{\sqrt{\prod_h^H 2\pi(\sigma_j^h)^2}} e^{-\sum_h^H \frac{(V_{ij}^h - \boldsymbol{\mu}_j^h)^2}{2(\sigma_j^h)^2}}$
  $\forall j \in \Omega_{L+1}$: $\boldsymbol{R}_{ij} \leftarrow \frac{\boldsymbol{a}_j p_j}{\sum_{u \in \Omega_{L+1}} \boldsymbol{a}_u p_u}$

Figure 1: EM-Routing Algorithm

Spread Loss Function

Spread loss function was proposed in paper [1]:

$$L_i = (\max(0, m - (a_t - a_i)))^2, L = \sum_{i \neq t} L_i \tag{1}$$

Capsule Visualization

We wonder if we can visualize what capsules learnt as we usually do for general CNN. Following is our results:

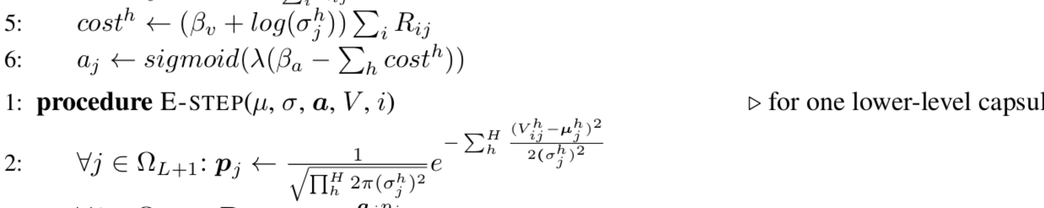

Figure 2: Visualizing the 32 primary capsules.

Difficulty of Matrix Capsule Implementation

We managed to implement matrix capsule since all implementation we can found seen to be untrustable. There are three main difficulty:

1. Matrix weight sharing for votes in kernel.
2. Broadcast matmul between transform matrix of shape [batch, kernel_size^2, channels_in, pose_height, pose_height] and [batch, out_img_height, out_img_width, kernel_size^2, channels_in, pose_height, pose_width].
3. Renorm routing weight r in the E-step.

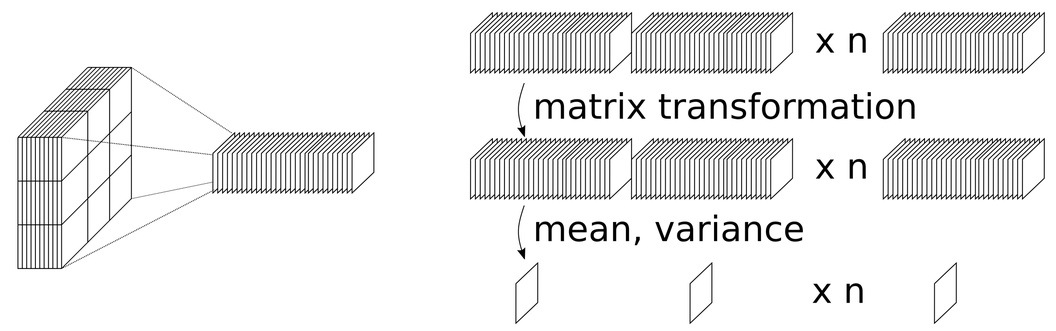

Figure 3: tf.extract_image_patches and M-step

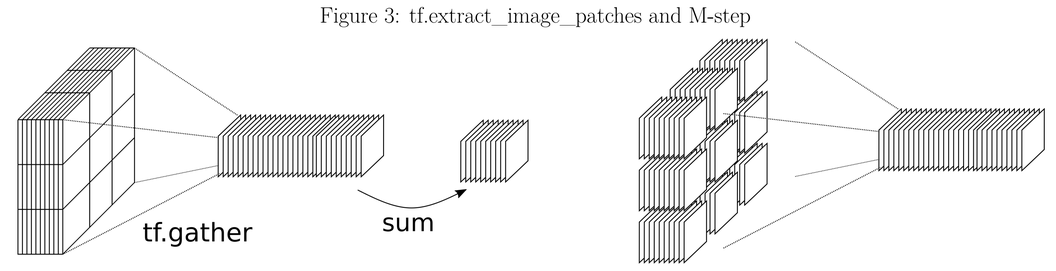

Figure 4: Calculation of $\sum_j a'_j p_j$

Matrix CNN

We think that using matrix transformation may have effect like kind of regularization. That is, consider CNN with input, output channels number equal to n^2, kernels size $k \times k$, there are $n^4 \times k \times k$ parameters to learn. But if matrix operation is used, only $n^2 \times n_{out} \times k \times k$. So experiment on using matrix transformation to generate upper later activation. That is,

$$A_{i,j}^{L+1} = \sum_{0 \leq k_i < k, 0 \leq k_j < k} M_{k_i,k_j} A_{i+k_i,i+k_j}^{L} \tag{2}$$

where $A_{i,j}^L, A_{i,j}^{L+1}, M_{k_i,k_j} \in \mathbb{R}^{n \times n}$.

	General CNN	Matrix CNN
Valid Accuracy	88.64%	89.25%
Testing Accuracy	83.68%	84.38%

Table 2: Accuracy of matrix CNN compared with general CNN.

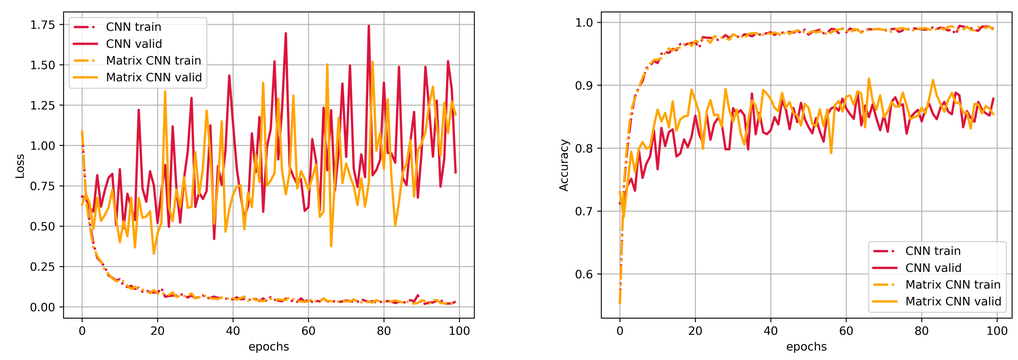

Figure 5: Loss (left) and accuracy (right) of matrix CNN compared with general CNN.

Robustness Against Adversarial Attack

Paper [1] claimed that their "Matrix Capsule with EM Routing" is more robust against adversarial attack, which we suspect is not because of its routing mechanism but its spread loss (equation (1)). Thus we experimented both FGSM and BIM on general CNN trained with spread loss. The results is as follow:

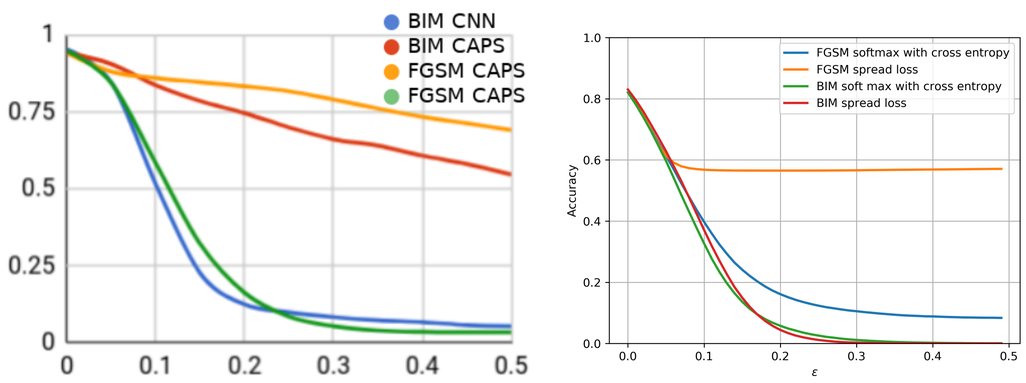

Figure 6: Accuracy against ϵ after an adversarial attack. Left is figure from paper [1]. Right is the result on general CNN.

Conclusion

In conclusion, we can see that some designs, like matrix transformation and spread loss is useful even for general CNN. However, routing seems to be necessary to get the performance claimed by the papers. The success of capsule is not solely because of matrix transformation or special loss.

References

[1] Anonymous. Matrix capsules with em routing. International Conference on Learning Representations, 2018.

[2] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. CoRR, abs/1710.09829, 2017.