LEARNING COMPOSITIONAL STRUCTURES FOR DEEP LEARNING: WHY ROUTING-BY-AGREEMENT IS NECESSARY

Anonymous authors
Paper under double-blind review

ABSTRACT

A formal description of the compositionality of neural networks is associated directly with the formal grammar-structure of the objects it seeks to represent. This formal grammar-structure specifies the kind of components that make up an object, and also the configurations they are allowed to be in. In other words, objects can be described as a parse-tree of its components - a structure that can be seen as a candidate for building connection-patterns among neurons in neural networks. We present a formal grammar description of convolutional neural networks and capsule networks that shows how capsule networks can enforce such parse-tree structures, while CNNs do not. Specifically, we show that the entropy of routing coefficients in the dynamic routing algorithm controls this ability. Thus, we introduce the entropy of routing weights as a loss function for better compositionality among capsules. We show by experiments, on data with a compositional structure, that the use of this loss enables capsule networks to better detect changes in compositionality. Our experiments show that as the entropy of the routing weights increases, the ability to detect changes in compositionality reduces. We see that, without routing, capsule networks perform similar to convolutional neural networks in that both these models perform badly at detecting changes in compositionality. Our results indicate that routing is an important part of capsule networks - effectively answering recent work that has questioned its necessity. We also, by experiments on CIFAR-10 and FashionMNIST, show that this loss keeps the accuracy of capsule network models comparable to models that do not use it .

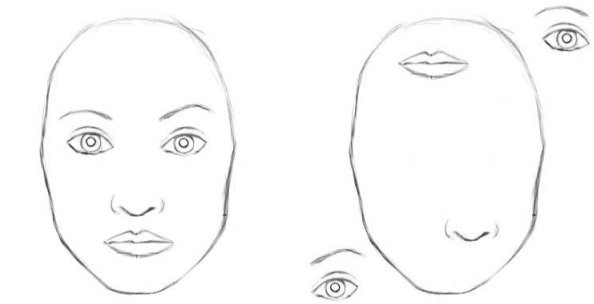
1 INTRODUCTION

The compositionality of visual objects can be described as being a parse-tree of objects, where objects at a level of this tree combine to form the object they are connected to at the higher-level. The types of objects, and their allowed configurations for combination can be described by category-specific rules. Therefore, the compositionality of an object can also be described by a spatial grammar, such that the derivation-rules of this grammar describe the parse-tree of the object.

This presents a candidate for the kind of connection-patterns among activations we wish to achieve in compositional neural networks - the representation of an object must be given by a tree-like set of activations across multiple layers. Each connection between activations of two layers can be thought of as representing a part-whole relationship between the objects they represent, and therefore a learned derivation-rule of some spatial grammar. In a setting with several compositionalities among inputs, this tree-like structure means that an activation in a layer of a neural network can be strongly connected to only one activation of a deeper layer.

What would be necessary, for such a structure, is a mechanism that decides what importance a derivation-rule is assigned. In our work, we identify that no such mechanism exists for convolutional neural networks (CNNs), while capsule networks (Sabour et al., 2017), (Hinton et al., 2018), (Venkataraman et al., 2020) can decide the importances of derivation-rules by their process of building deeper activations.

Figure 1: Example of a change in compositionality in face images. The image on the left shows the compositionality of a face. The image on the right has the same parts, but not the same relationships between them. Images are from (fac, 2010 (accessed October 1st, 2020))



We briefly discuss this procedure. Deeper capsules are built from a consensus-based aggregation of predictions made for them by shallower capsules. By having capsules activated only when predictions for them are in agreement, this procedure allows for a checking of whether the shallower capsules share a common viewpoint and can form the corresponding object. This process is termed routing.

A class of routing algorithms, for example, dynamic routing (Sabour et al., 2017) aggregates predictions by a weighted-sum, each weight denoting the extent of the consensus of a prediction for a deeper capsule relative to the consensus of the predictions of the corresponding capsule for other deeper capsules. For a parse-tree structure, only one routing-weight for a shallower capsule must be strong. We identify that a low entropy of routing weights for each capsule can improve the compositional structure of capsule networks.

However, this does not happen in the dynamic routing where we show that the entropy of routing weights is high. To remedy this, we minimize entropy of routing weights, and subsequently demonstrate that we are able to achieve better compositional representations. A measure of this can be seen in the ability to detect changes in compositional structures. Figure 1 shows an example of such a change. We can measure a network's compositionality by observing its behaviour on such data. We show that CNNs do not detect differences between such data, while capsule networks can. We show that capsule networks with a low entropy of routing weights detect such differences more than capsule networks with a higher entropy of routing weights. This work, to our knowledge, is the first study of the compositional structure of capsule networks and routing.

We also show that entropy-regularised capsule networks achieve similar performance to unregularised networks on classification tasks on CIFAR-10 and FashionMNIST. We list our contributions below:

1. A formal language description of CNNs and capsule networks to show their difference in compositionalities. (see section 3)
2. Using entropy of the routing weights as a loss function for better compositionality. (see section 4)
3. Experiments on capsule networks that show that entropy-regularised capsule networks are able to detect changes in compositional structures. Unregularised capsule networks do not show the same extent of this ability, and CNNs do not show it. (see section 5.1)
4. Experiments on CIFAR10 and FashionMNIST that show entropy-regularised capsule networks have similar accuracy to other capsule networks. (see section 5.2)

2 Related work

Grammar-models are widely used in computer vision, for example, (Zhu & Mumford, 2007), (Tu, 2015), (Lin et al., 2009), (Zhao & Zhu, 2011). These models learn a grammar-structure directly for

representing the compositionality of objects, and use the learned grammar in computer vision tasks. There are several issues with directly implementing grammar models such as in (Zhu & Mumford, 2007); for example, the need for annotations of parts. Further problems are the difficulty in scaling to deeper structures and relatively lower performance - as discussed in (Tang et al., 2017). One means of having high-performing grammatical models is to use grammar-concepts in deep learning. Some work in this direction can be seen in Li et al. (2019) and Tang et al. (2017). However, these models are not a parse-tree representation, and do not incorporate any notion of choosing between spatial relationships among objects, which is what we wish to do.

An alternative approach that is not as grounded in formal-language theory, but attempts to build tree-like representations for data is the capsule network model. Several capsule network models exist that detail various routing algorithms and architectures: (Sabour et al., 2017), Hinton et al. (2018), (Wang & Liu, 2018), (Ahmed & Torresani, 2019), (Rajasegaran et al., 2019), (Venkataraman et al., 2020), and Choi et al. (2019). However, these and many other models that we have not mentioned do not address the issue of analysing part-whole relationships. A recent work by (Peer et al., 2018) works on building a new routing algorithm that reduces the entropy of routing-weights. However, their paper does not test the compositionality of capsule networks - and, as we show, does not reduce the entropy to a small value. (Kissner & Mayer, 2019) proposes a grammar-structure based on units termed capsules. This work does not, however, propose a means of studying the compositionality of models, and does not learn compositional structures from unannotated inputs.

3 Formal Language Description of CNNs and Capsule Networks

In this section, we formally describe a grammar-structure for CNN-models and capsule networks. We begin by doing so for CNNs. Before we present a grammar-description of CNN-models, we first present definitions for the layers that our grammar describes.

3.1 CNN-models

We consider CNN-models as functions on groups so that our work extends to group-equivariant CNNs (Cohen & Welling, 2016). The following definitions follow from the same work.

Correlation: Consider a group $(G, .)$, functions $f : G \times (\mathbb{N} \cup \{0\}) \to \mathbb{R}$ and $\Psi : G \times (\mathbb{N} \cup \{0\}) \times (\mathbb{N} \cup \{0\}) \to \mathbb{R}$. Then, the correlation operator is given by $(f \star \Psi)(g, i) = \Sigma_{h \in G} \Sigma_{k=0}^{d-1} (f(h, k)\Psi(g^{-1}.h, k, i))$ where d is the number of inchannels.

Activation function: Consider a group $(G, .)$, and a function $f : G \times (\mathbb{N} \cup \{0\}) \to \mathbb{R}$. Let $v : \mathbb{R} \to \mathbb{R}$. Then $C_v f(g, n) = v(f(g, n))$ represents the point-wise activation function.

Max-pooling: As in (Cohen & Welling, 2016), we split max-pooling into two steps, pooling and subsampling. Consider a group $(G, .)$. Let $f : G \times \mathbb{N} \cup \{0\} \to \mathbb{R}$. Further, let U be a subset of G. Then, the pooling step is defined by $MaxPool f(g, i) = max_{k \in g.U} f(k)$. The subsampling step, which is used in the case of strides, is given by subsampling the maxpooled function on a subgroup of G.

We consider a layer of such a model to be a correlation-layer followed by an activation function, or a max-pooling layer, or layer resulting in a weighted sum of two layers. The grammar for such a model can be described as follows.

We define a grammar defined on a group $(G, .)$ as the tuple (Σ, N, S, R, f), where:

Σ is the set of terminals. These denote patterns that do not have parts.

N is the set of non-terminals. These denote patterns in the intermediate layers of the compositionality.

S is the start symbol. This denotes a complete pattern.

R is the set of derivation-rules. Each derivation-rule can be one of two types:

1 $X_1|X_2...|X_n \to C$, where $X_1, X_2, ..., X_n \in N$, and C is a sequence of terminals and non-terminals.

2 $X \rightarrow C_1 C_2 ... C_n$, where $C_1, C_2, ..., C_n \in \Sigma \cup N$, and $X \in \{C_1, C_2, ..., C_n\}$.

f is a function $f : N \times G \times \mathbb{N} \cup \{0\} \rightarrow \mathbb{R}$. f denotes the activation of an instance of a non-terminal.

CNNs are an example of this grammar. The input is treated as a sequence of terminals. Each activation is treated as an instance of a non-terminal. Further, each process of obtaining a deeper activation is treated as an application of a derivation-rule. Two types of rules exist. The first type can be described as an enumeration of alternate derivations from a set of non-terminals. This corresponds to the non-pooling layers of the CNN, where a local pool of activations is combined using multiple convolutions to obtain more than one activation, in general.

The second is a subsampling rule, that chooses one non-terminal or terminal from a set of terminals and non-terminals. This represents the max-pooling layers.

The activation function in the grammar gives the actual value of the activation associated with each instance of a non-terminal.

3.2 CAPSULE NETWORKS

Capsule networks are a recent family of neural networks. Each activation of a capsule network is a vector that represents the pose of an instance of an object of some type. Deeper capsules are built through routing by combining predictions made by shallower capsules for them in a consensus-based manner. Routing aims to capture the level of agreement in viewpoints among components.

Several capsule models exist, with multiple prediction strategies and routing algorithms (Sabour et al., 2017), (Venkataraman et al., 2020), (Rajasegaran et al., 2019), (Hinton et al., 2018). In our work, we use SOVNET model Venkataraman et al. (2020) with dynamic routing.

The SOVNET model associates each capsule-type with a neural network defined on a group $(G, .)$. Shallower capsules use this neural network to make predictions for deeper capsules. The procedure for using dynamic routing while using convolutional predictions is given in Algorithm 1.

Algorithm 1 The dynamic routing algorithm for SOVNET

Input: $\{f_i^l | i \in \{0, ..., N_l - 1\}, f_i^l : G \rightarrow \mathbb{R}^{d^l}\}$ where N_l is number of capsules in layer l

Output: $\{f_j^{l+1} | j \in \{0, ..., N_{l+1} - 1\}, f_j^{l+1} : G \rightarrow \mathbb{R}^{d^{l+1}}\}$

Trainable Functions: $(\Psi_j^{l+1}, \star), 0 \leq j \leq N_{l+1} - 1$, - a set of d^{l+1} group-equivariant convolutional filters (per capsule-type) that use the group-equivariant correlation operator $\star$

$S_{ijp}^{l+1}(g) = (f_i^l \star \Psi_j^{l+1,p})(g) = \sum_{h \in G} \sum_{k=0}^{d^l - 1} f_{ik}^l(h) \Psi_k^{l+1,p}(g^{-1} \circ h); p \in \{0, ..., d^{l+1} - 1\}$

$b_{ij}^{l+1}(g) = 0 \; \forall 0 \leq i \leq N_l - 1, \forall 0 \leq j \leq N_{l+1} - 1, \forall g \in G$

for iter in ITER **do**

$c_{ij}^{l+1}(g) \leftarrow \frac{exp(b_{ij}^{l+1}(g))}{\sum_{k=0}^{N_{l+1}-1} exp(b_{ik}^{l+1}(g))}$

$f_j^{l+1}(g) = \sum_{i=0}^{N_l - 1} c_{ij}^{l+1}(g) S_{ij}^{l+1}(g) \; \forall \; 0 \leq j \leq N_{l+1} - 1, \forall g \in G$

$f_j^{l+1}(g) = Squash(f_j^{l+1}(g)) = \frac{\|f_j^{l+1}(g)\|}{1 + \|f_j^{l+1}(g)\|^2} \; f_j^{l+1}(g); \forall \; 0 \leq j \leq N_l - 1, \forall \; g \in G$

$b_{ij}^g = b_{ij}^{l+1}(g) + S_{ij}^{l+1}(g).f_j^{l+1}(g), \forall g \in G, \forall 0 \leq i \leq N_l - 1, \forall 0 \leq j \leq N_{l+1} - 1$

We now describe this capsule network model in terms of a grammar. Consider the grammar defined on a group $(G, .)$, and given by the tuple (Σ, N, S, R, f), where

Σ is the set of terminals.

N is the set of non-terminals.

S is the start symbol, and represents a complete pattern.

R is the set of derivation-rules. Derivation-rules are of two types.

OR-rule $X_1|X_2...|X_n \rightarrow C$, where $X_1, X_2, ..., X_n \in N$, and C is a sequence of terminals and non-terminals. Further, there exists a likelihood-function p that maps each derivation $X_i \rightarrow C$ to a number between 0 and 1, such that $\Sigma_{i=0}^{n} p(X_i) = 1$. We term $X_1, X_2, ..., X_n$ as OR-nodes.

AND-rule $X \rightarrow C_1 C_2 ... C_I$, where $X \in N$, and $C_1, C_2, ..., C_I$ are OR-nodes. X is termed an AND-node.

f is a function $f : N \times G \times \mathbb{N} \cup \{0\} \rightarrow \mathbb{R}^d$. f represents the activation of an instance of a non-terminal.

Capsule networks as given in Algorithm 1 are an example of this grammar. The terminals denote the input. Each prediction made by a capsule for a deeper capsule is an OR-rule. The process of assigning routing-weights is the likelihood function. The AND-rule combines predictions to form the deeper capsules.

4 ENTROPY AMONG DERIVATIONS

In order to have a parse-tree structure of active derivations, the likelihood function for an OR-rule must have only one large value. This corresponds to a low entropy. In the case where the entropy is large, there is little difference between CNNs and capsule networks in terms of compositionality. This is because in such a case, there is little difference in the importance among alternate derivations.

The entropy of the routing-weights of capsule networks with dynamic routing, however, are not always low (Peer et al., 2018). In which case, to have a compositional representation, the entropy must be reduced. We propose a loss function to this end.

We use the sum of the mean entropies of the routing-weights $c_{ij}^{l}(g)$ for each layer of the network. Thus, in addition to a loss function for the task, we also use a loss for compositionality. This is an alternate approach to the work of Peer et al. (2018), where a routing algorithm, termed scaled-distance-agreement routing, was used to reduce the entropy of routing-weights.

5 EXPERIMENTS

We conduct two sets of experiments to test the compositionality and accuracy of entropy-regularised capsule networks. The first experiment involves training models on data with a compositional structure and testing it on data with a different compositionality. The second experiment involves checking the accuracy of models on CIFAR10 and FashionMNIST. We describe each of these experiments in more detail in the following sections.

5.1 DETECTING CHANGES IN COMPOSITIONAL STRUCTURES

Since we are interested in testing the ability of models to detect changes in compositional structure, we constructed a dataset where this is possible. We used the task of determining if an input is a face-image or not as the learning-task. We randomly selected 50,000 images of the MORPH dataset - each image frontalised and aligned - for the train-set of faces, and used the train-set - of 50,000 images - of the Animal-10N dataset for non-face images. The images of the MORPH dataset(Rawls & Ricanek, 2009) and the test-set of the Animal-10N dataset (Song et al., 2019) formed the test-set. The images were resized to 128 pixels in height and width, and were randomly flipped in the horizontal direction at train-time.

We trained four CNN-models on this data. The first CNN-model and the second CNN-model have the same architecture, but use different losses. The first CNN used the margin-loss (Sabour et al., 2017) for its predictions. The second CNN used the cross-entropy loss. The architecture used for these two models uses regular correlations in its layers. Further, the architecture also used max-pooling. The third model is based on the Resnet architecture, and used the margin loss for its predictions. The fourth architecture used group-equivariant convolutions (Cohen & Welling, 2016), and used the margin loss for its predictions. The exact details of the models and their training can

be found in our code which avaliable `https://github.com/compositionalcapsules/entropy_regularised_capsule`. We present the accuracy obtained from the models at the last epoch of training in Table 1.

Table 1: Accuracies of four CNN-models on the face vs. non-face task, averaged over three runs.

Method	Mean	Standard deviation
CNN1	100.00%	0.00%
CNN2	100.00%	0.00%
CNN3	99.93%	0.00%
CNN4	100.00%	0.00%

We further trained several capsule network models on this task. The capsule networks are based on two architectures, the specifications of which are available at `https://github.com/compositionalcapsules/entropy_regularised_capsule`. Each model is under a different amount of regularisation. They all use margin loss as the classification loss. The entropy-loss is added to this loss in a weighted-sum. The weight gives the amount of the regularisation. One unregularised model and a regularised according to a schedule, based on the first architecture are trained for this task. The schedule slowly increases the weight of the entropy-loss, while reducing the weight of the margin loss as the training continues. We term these models, unregcaps1 and schcaps1 to denote the unregularised model and the model regularised according to a schedule.

Table 2: Accuracies and entropies of capsule network models on the face vs. non-face task, averaged over three runs.

Method	Mean of accuracy	Standard deviation of accuracy	Mean of entropy	Standard deviation of entropy
Unregcaps1	99.96%	0.01%	2.29	0.01
schcaps1	99.44%	0.08%	0.004	0.001
equalcaps1	99.96%	0.00%	5.0	5.0
sdacaps1	99.98%	0.01%	2.73	0.01
unregcaps2	99.99%	0.01%	2.09	0.05
0.4caps2	96.63%	2.03%	0.012	0.008
0.8caps2	99.29%	0.59%	0.0002	0.0001
schcaps2	99.93%	0.07%	0.005	0.004
equalcaps2	99.99%	0.00%	5.0	5.0
sdacaps2				

For the second architecture, we train four models. These are an unregularised model, a model regularised by weighting the entropy-loss by 0.4 and the margin-loss by 0.6, a model regularised by weighting the entropy-loss by 0.8 and the margin-loss by 0.2 and a model regularised according to a schedule that increases the weight of the entropy-loss while reducing the weight of the margin-loss as the training continues. We use the terms unregcaps2, 0.4caps2, 0.8caps2, schcaps2 to denote these models.

We also trained capsule network models that give equal routing-weights to all the predictions, and models that uses the scaled-distance-agreement routing algorithm in (Peer et al., 2018). We use equalcaps1, equalcaps2, sdacaps1, and sdacaps2 to denote these models. The code for the architectures and the training is given in our code. The accuracies and the mean of the entropy-loss for the test-set are given in Table 2.

All the models, CNNs and capsule networks, performed very well in this task. However, 0.4caps2 is not as high-performing and has a large standard deviation in its results. It still is able to classify with above 96% accuracy, and can be considered a good model for classification. We used the trained models in these tasks to detect changes in compositionality of the face images.

Specifically, we modify the face-images of the test-set by interchanging parts of the face. We interchange among the eyebrows, the eyes, the nose and the mouth regions. We do so by detecting the landmark points for the face, and then finding the regions based on this. The regions are resized

while being changed as they could be of different sizes. The modified test-images thus are from face-images, but do not have the same compositionality.

A model that is compositionalwill not be as activated on such images as much as they would be on faces. Thus, the mean activation for the predictions will be lower for compositional models on these images. The mean of the activations for the test-data is given in Table 3.

Table 3: Mean activations of the predictions for face data

Model	Mean activation for faces	Mean activation for transformed faces	Model	Mean activation for faces	Mean activation for transformed faces
CNN1	0.633	0.633	sdacaps1	0.730	0.632
CNN2	1.00	0.984	unregcaps2	0.730%	0.679
CNN3	0.632	0.592	0.4caps2	0.714	0.391
CNN4	0.629	0.628	0.8caps2	0.725	0.4836
Unregcaps1	0.730	0.646	schcaps2	0.729	0.591
schcaps1	0.727	0.546	equalcaps2	0.729	0.719
equalcaps1	0.727	0.714	sdacaps2		

This experiment is an answer to papers (Paik et al., 2019), (Gu & Tresp, 2020) that show that routing algorithms are not required for performance. While deep learning models for computer vision do not need a routing-like mechanism for achieving high performance, we see that such models need not learn the compositionality of their inputs. The CNN-models perform very poorly in this regard, showing no change in the activations. This is both in models that have max-pooling, and those which do not use it. Max-pooling is thought to reduce the ability of CNNs to learn spatial relationships due to the fact that the position of the original activation is not preserved, and this could cause a loss of spatial relationships among activations. We see that even when no pooling is used, CNNs do not detect learn compositional structures well.

We see that the performance of capsule networks, in detecting changes in compositionality, improves with a lower entropy among routing-weights. The capsule network model that gives equal weight to each routing-weight performs similarly to a CNN. We also note that the architecture of the model is also important to learning accurate compositional representations; the entropy among routing-weights is a measure of the tree-like nature of the representation, the learned derivation-rules depend on the architecture. Our experiments show that accuracy of a model in tasks is not proof of a good representation. For computer vision, one aspect that must be considered is the compositionality of the model.

5.2 Experiments on classification

The second experiment that we conduct is to test the classification performance of entropy-regularised models. We trained and tested three sets of models on CIFAR10 and FashionMNIST.

Table 4: Accuracies and entropies of unregularised and regularised capsule network models on CIFAR10, averaged over three runs.

Method	Mean of accuracy	Standard deviation of accuracy	Mean of entropy-loss	Standard deviation of entropy-loss
unregcaps1	91.48%	0.09%	3.47	0.07
0.4caps1	90.46%	0.29%	0.0004	$9.210\times^{-5}$
0.8caps1	90.31%	0.69%		
sch1caps1	99.63%	0.05%		
sdacaps1				

Method	Mean of accuracy	Standard deviation of accuracy	Mean of entropy-loss	Standard deviation of entropy-loss
unregcaps2	91.09%	0.12%	8.98	0.01
0.4caps2	90.50%	0.27%	0.044	0.0008
0.8caps2	73.68%	0.31%	0.021	0.00074
sch1caps2	90.91%	0.08%	0.29	0.0011
sdacaps2	89.23%	0.13%	5.19	0.02

Method	Mean of accuracy	Standard deviation of accuracy	Mean of entropy-loss	Standard deviation of entropy-loss
unregcaps3	93.65%	0.17%	1.87	0.03
0.4caps3	93.14%	0.16%	0.0042	0.00079
0.8caps3	92.37%	1.25%	0.00037	0.00034
sch1caps3	91.02%	0.05%	0.035	0.008
sdacaps3	85.47	0.37	2.06	0.005

Each model is trained under different combinations of the classification-loss and entropy-regularisation. The regularisation is done by a weighted-sum of the entropy-loss and the classification-loss. The weights are fixed or changed according to a schedule. We use a weight of 0.6 for the entropy-loss and 0.4 for the classification-loss in one case and 0.8 for the entropy-loss and 0.4 for the classification-loss in the other. We also train with one schedule which reduces the

Table 5: Accuracies and entropies of unregularised and regularised capsule network models on FashionMNIST, averaged over three runs.

Method	Mean of accuracy	Standard deviation of accuracy	Mean of entropy-loss	Standard deviation of entropy-loss
unregcaps1	92.73%	0.01%		
0.4caps1				
0.8caps1				
sch1caps2				
sdacaps1	93.02	0.10	0.09	0.0007

Method	Mean of accuracy	Standard deviation of accuracy	Mean of entropy-loss	Standard deviation of entropy-loss
unregcaps2	92.73%	0.01%	1.5	0.01
0.4caps2				
0.8caps2				
sch1caps2				
sdacaps1				

Method	Mean of accuracy	Standard deviation of accuracy	Mean of entropy-loss	Standard deviation of entropy-loss
unregcaps3	93.98%	0.01%		
0.4caps3				
0.8caps3				
sch1caps3				
sdacaps3				

weight of the classification-loss and increases the weight of the entropy-loss as the training continues.We further train a model that uses the routing algorithm in Peer et al. (2018). The naming conventions of these models follow from section 5.1.

The accuracy of the models is given in the Tables 4 and 5. The mean entropy-loss for the test-set for the models is given along with the accuracies. In general, the entropy-regularised models perform similarily to the unregularised models and the models which use the routing algorithm in Peer et al. (2018). One observation is that the results of entropy regularisation can lead to variations in accuracy across runs - reflecting the fact that optimising with a loss for every layer is not easy. However, our goal here is to show that under identical training conditions, models which are compositional perform on par with unregularised models in terms of accuracy. We did not aim for state-of-the-art results which may be possible with explicit hyperparameter tuning.

6 Conclusion

We present a grammar-based framework for describing the compositionality of CNNs and capsule networks. We show that CNN-models do not have a compositional structure as they do not have a means of assigning importances to derivation-rules. We show that capsule networks with dynamic routing can activate derivation-rules based on the part-whole relations of the input.

We present an entropy-loss for improving the ability of capsule networks to detect compositional structures in data. We show, by experiments on face-data, that this entropy-loss improves the compositional-structure of capsule networks. Capsule networks that use the entropy-loss are able to better detect changes in compositional structure than unregularised capsule networks, while CNN-models perform poorly in this. We also see that capsule networks that use this entropy-loss perform similarily to unregularised capsule models.

In order to improve the performance of capsule networks whose entropy among routing-weights is low, we believe that routing algorithms that ensure a low entropy must be developed, as also methods to reduce the effect of background. Another area that could improve the performance of these models would be developing architectural units for better training.

References

2010 (accessed October 1st, 2020). URL `http://sharenoesis.com/wp-content/uploads/2010/05/7ShapeFaceRemoveGuides.jpg`.

Karim Ahmed and Lorenzo Torresani. Star-caps: Capsule networks with straight-through attentive routing. In *Advances in Neural Information Processing Systems*, pp. 9101–9110, 2019.

Jaewoong Choi, Hyun Seo, Suii Im, and Myungjoo Kang. Attention routing between capsules. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999, 2016.

Jindong Gu and Volker Tresp. Improving the robustness of capsule networks to image affine transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7285–7293, 2020.

Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *International conference on learning representations*, 2018.

Michael Kissner and Helmut Mayer. A neural-symbolic architecture for inverse graphics improved by lifelong meta-learning. In *German Conference on Pattern Recognition*, pp. 471–484. Springer, 2019.

Xilai Li, Xi Song, and Tianfu Wu. Aognets: Compositional grammatical architectures for deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6220–6230, 2019.

Liang Lin, Tianfu Wu, Jake Porway, and Zijian Xu. A stochastic graph grammar for compositional object representation and recognition. *Pattern Recognition*, 42(7):1297–1307, 2009.

Inyoung Paik, Taeyeong Kwak, and Injung Kim. Capsule networks need an improved routing algorithm. *arXiv preprint arXiv:1907.13327*, 2019.

David Peer, Sebastian Stabinger, and Antonio Rodriguez-Sanchez. Increasing the adversarial robustness and explainability of capsule networks with γ-capsules. *arXiv preprint arXiv:1812.09707*, 2018.

Jathushan Rajasegaran, Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Suranga Seneviratne, and Ranga Rodrigo. Deepcaps: Going deeper with capsule networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10725–10733, 2019.

Allen W Rawls and Karl Ricanek. Morph: Development and optimization of a longitudinal age progression database. In *European Workshop on Biometrics and Identity Management*, pp. 17–24. Springer, 2009.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pp. 3856–3866, 2017.

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pp. 5907–5915, 2019.

Wei Tang, Pei Yu, Jiahuan Zhou, and Ying Wu. Towards a unified compositional model for visual pattern modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2784–2793, 2017.

Kewei Tu. Stochastic and-or grammars: A unified framework and logic perspective. *arXiv preprint arXiv:1506.00858*, 2015.

Sai Raam Venkataraman, S Balasubramanian, and R Raghunatha Sarma. Building deep equivariant capsule networks. In *International Conference on Learning Representations*, 2020.

Dilin Wang and Qiang Liu. An optimization view on dynamic routing between capsules. 2018.

Yibiao Zhao and Song-Chun Zhu. Image parsing with stochastic scene grammar. In *Advances in Neural Information Processing Systems*, pp. 73–81, 2011.

Song-Chun Zhu and David Mumford. *A stochastic grammar of images*. Now Publishers Inc, 2007.