MULTITASK LEARNING WITH CAPSULE NETWORKS FOR SPEECH-TO-INTENT APPLICATIONS

Jakob Poncelet , Hugo Van hamme

KU Leuven
Department Electrical Engineering ESAT-PSI
Kasteelpark Arenberg 10, Bus 2441, B-3001 Leuven Belgium
{jakob.poncelet, hugo.vanhamme}@esat.kuleuven.be

ABSTRACT

Voice controlled applications can be a great aid to society, especially for physically challenged people. However this requires robustness to all kinds of variations in speech. A spoken language understanding system that learns from interaction with and demonstrations from the user, allows the use of such a system in different settings and for different types of speech, even for deviant or impaired speech, while also allowing the user to choose a phrasing. The user gives a command and enters its intent through an interface, after which the model learns to map the speech directly to the right action. Since the effort of the user should be as low as possible, capsule networks have drawn interest due to potentially needing little training data compared to deeper neural networks. In this paper, we show how capsules can incorporate multitask learning, which often can improve the performance of a model when the task is difficult. The basic capsule network will be expanded with a regularisation to create more structure in its output: it learns to identify the speaker of the utterance by forcing the required information into the capsule vectors. To this end we move from a speaker dependent to a speaker independent setting.

Index Terms— Spoken Language Understanding, Capsule Networks, Multitask Learning, End-to-end, Speaker Identification

1. INTRODUCTION

Technology is advancing at an unprecedented rate, ultimately trying to ease and improve the life of people. Speech recognition is playing a major role in this trend to allow hands-free operation of all kinds of devices. Voice control using spoken language understanding (SLU) systems can be beneficial in all parts of daily life, but more specifically it would help physically challenged and elderly people to live independently. Command-and-control (C&C) applications are typical in this setting, e.g. for positioning of their bed, operating domestic devices, etc. However this requires the system to understand non-standard speech as well, like thick dialects or impaired speech, which is more frequent in these user groups. This is where common speech technology based on acoustic models runs into problems [1]. A speech-to-intent understanding system can be more robust to variations and errors in speech, since it doesn't use an intermediate textual representation, and is attracting more and more research interest [2, 3, 4].

In [5] an SLU system for C&C has been implemented, which builds up a model from scratch using demonstrations from the user. The system learns to map the spoken commands uttered by the user directly to a semantical representation with labels for every task (speech-to-intent). Building a model from scratch from user demonstrations, i.e. without making linguistic assumptions such as the phone set, vocabulary or grammar, makes it also accessible for deviant speech and multiple application and language domains. Moreover it allows the user to choose how to phrase the commands, instead of being confined to the wording chosen by the designer.

In this paper the implementation of the aforementioned SLU system, which is built with capsule networks, will be analysed and adapted. Capsule networks were presented in 2017 by Hinton [6] and are a new type of deep neural network (DNN), believed to need less training data than standard DNN's. The proposed capsule networks have been compared in accuracy and data requirements in this setting to a previously proposed Non-negative Matrix Factorisation (NMF) approach [7, 8]. The capsule network was deemed very promising, since it often outperformed the other architectures [9], hence an insight into its working would be useful.

A capsule network consists of layers of capsules, with each capsule being characterised by a vector (as compared to scalar neurons). The activation vectors of the capsules in the output layer are essentially a condensed representation of the information that the network uses to classify the speech to the right intent. The effect of the dimension of these vectors is examined to get an indication of its importance and the ben-

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

efit of giving the capsules more freedom for the orientation of the activation vectors. If a low-dimensional space would suffice without introducing errors, the number of parameters can be reduced to improve efficiency.

Besides allowing the capsules to freely put in their output whatever they like for the classification of an utterance, one can also force them to use the dimensions of the vectors to store information and create a more structured output. The network can be extended to learn auxiliary tasks at the same time because of these higher dimensional vectors. In other words, multitask learning can be implemented by applying regularisations to the output vectors, which could improve the efficiency and performance of the model [10, 11]. In this paper, the model will learn to identify the speaker that uttered the command by encapsulating information into the orientation of the capsule vectors. Learning which speaker gave the command is a useful task for the system to incorporate when decoding the utterances and might improve the performance [12]. Consequently we move from a speaker dependent setting (as is [8]) to a speaker independent setting. Training a model with mixed data from multiple speakers gives a penalty in learning speed, since different speakers use different phrasings for their commands and acoustic speaker variation needs to be learnt as well. From a practical point of view, speaker identification allows an SLU system to be shared by multiple users and the system would be able to independently figure out for which person a task has to be carried out (which could be different from person to person).

The basic and extended model will first be explained in section 2, along with some basic theory about capsule networks. Section 3 discusses specifics about the methods used in the experiments and section 4 describes the performed experiments. In section 5 the results are discussed and section 6 finally gives a conclusion to this work.

2. MODEL

2.1. Capsule Network Baseline Model

A capsule network consists of different layers of capsules. A capsule is characterised by an activation vector $\boldsymbol{u}_i$. The length of this vector corresponds to the probability of an object being present, and the orientation of this vector corresponds to the parameters of the object (for example the pose). Every capsule in a layer will try to predict the output of the capsules in the next layer. This prediction uses a transformation matrix $\boldsymbol{W}_{ij}$ for every capsule pair in consecutive layers that will be learned by backpropagation of the loss through the network.

$$\hat{\boldsymbol{u}}_{j|i} = \boldsymbol{W}_{ij}\boldsymbol{u}_i \tag{1}$$

The connection between two layers uses a dynamic routing algorithm, as explained in [6], and is based on agreement between predictions of the high level capsule property by the lower level capsules. After a few iterations the output vector of the capsules in the subsequent layer is obtained. Note that the length of this vector is between 0 and 1 using a squash function. Finally for classification purposes a margin loss can be implemented proceeding from the length of the activation vectors $\boldsymbol{v}_k$ of the last capsule layer. For K classes this is defined as in (2), with T_k equal to 1 if class k is present (and 0 otherwise), and $m^+ = 0.9$ and $m^- = 0.1$ chosen boundary values. Every output capsule corresponds to a task label and the decoded task is decided based on the most active capsules, i.e. with an activation vector with norm close to 1

$$L_l = \sum_{k=1}^{K} T_k \, max(0, m^+ - \|\boldsymbol{v}_k\|) + (1 - T_k) \, max(0, \|\boldsymbol{v}_k\| - m^-) \tag{2}$$

A more detailed explanation about capsule networks can be found in [6]. The implementation in [5, 9] with two layers (a primary capsule layer and an output capsule layer) serves as the baseline model that will be used for the experiments in this paper.

2.2. Multitasking Model with Speaker Identification

We want to give a meaning to the dimensions of the output capsules through multitask learning, so the model will use them more actively and create a more structured output. This was previously not explicitly required of the capsules, since the classification is only based on the length of the output vectors (as in (2)), not on the orientation. In this section the baseline model is extended with an additional layer to learn which speaker spoke the command.

We start with a definition of the average capsule $\boldsymbol{z}$ in each utterance, with N the number of output capsules.

$$\boldsymbol{z} = \frac{\sum_{i=1}^{N} \boldsymbol{v}_i}{\sum_{i=1}^{N} \|\boldsymbol{v}_i\|} \tag{3}$$

The average capsule combines for every output dimension the information of the vectors of all output capsules (it averages over them). The average capsule is thus a column vector of dimension equal to the dimension of the output capsules, for example 8. A single-layer neural network followed by a softmax layer will map the average capsules to speaker probabilities. The weight matrix of this layer will be called the projection matrix $\boldsymbol{W_s}$ and has dimensions $(n \times M)$, with n the output dimension of the capsules and M the number of speakers in the dataset. In the testing phase, the model chooses the speaker with the highest probability. A schematic of the multitask model is shown on Fig. 1.

To let the model learn, a new speaker loss term is added to the total loss (for now only consisting of the label loss as in (2)). The speaker loss uses a cross-entropy loss function based on the target speaker (as a one-hot encoded vector $\boldsymbol{t}$) and the estimated probabilities P_i for every speaker i, and is summed over all M speakers.

$$L_s = -\sum_{i=1}^{M} t_i log(P_i) \tag{4}$$

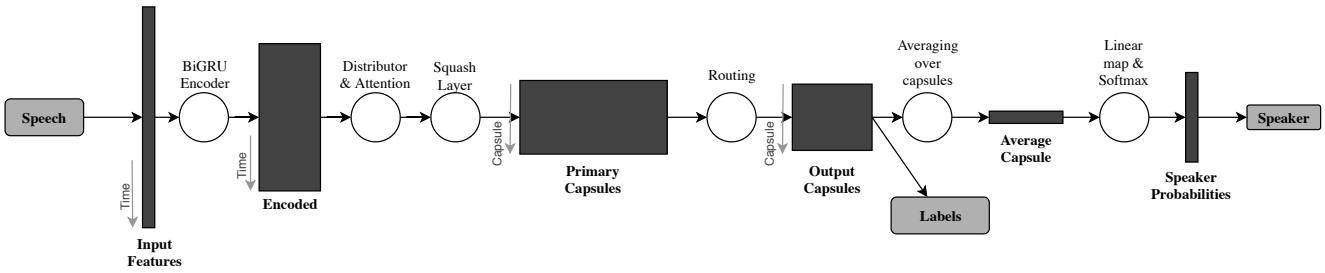


Fig. 1: Schematic of the multitask model

Backpropagation of the loss will then adapt the trainable matrix $\boldsymbol{W_s}$ to correctly identify the speaker and will force the capsules to encapsulate this into their orientation. A regularisation parameter λ_s, defined as the speaker weight, is added in the total loss function to weigh the relevance of the speaker loss compared to the label loss.

$$L_{tot} = L_l + \lambda_s L_s \tag{5}$$

3. METHOD

3.1. Dataset

The model will be tested on two publicly available datasets. The GRABO dataset [13, 14] is based on a setting where a person gives commands to a robot. The robot can move around, pick things up and point a laser. There are a total of 33 different output labels corresponding to possible positions, movement speeds and actions. Data has been recorded from ten Dutch speakers and one English speaker. With around 6000 recorded utterances, this is a smaller dataset with little variety, e.g. most commands have the same structure of sentences.

The Fluent Speech Commands dataset by Fluent.ai [15, 16] is a larger and more challenging dataset. It comprises 30000 utterances from 97 speakers, used in a smart-home controlling appliance setting, for e.g. controlling the lights or music volume in a certain room. There are 31 unique intent labels, but there is much more variation in the spoken commands. We should point out that there are some speakers with only a few recorded utterances.

3.2. Experimental Setup

Most of the experiments in this paper are cross-validation experiments. The dataset is divided into 150 blocks. Starting from one block, the model will be trained on an increasing number of blocks, and tested on all remaining blocks. This way a learning curve is created. In the speaker dependent experiments with the baseline model, the data is fed to the model speaker by speaker and the final curve is obtained by averaging over the results for every speaker. On the contrary, in the (speaker independent) experiments involving speaker identification, the utterances from all speakers are randomly shuffled beforehand and then all data is divided into blocks.

The hyperparameters of the model are chosen as in [5] and are not altered, except when specifically mentioned. The varying parameters will be the dimension of the output capsules and the regularisation weight of the multitask model.

Evaluation of the label classification task is done using an F1 score [17]. To evaluate the identification of the speakers, we use the percentage of correctly decoded speakers.

4. EXPERIMENTS

First of all the effect of the dimension of the output capsules was analysed by comparing experiments for different output dimensions (ranging from 2 to 8). The analysis was done with the baseline model on the GRABO dataset in a speaker dependent setting. We observed that the output capsule dimension had little to no impact on the F1 scores, even down to a capsule dimension of 2.

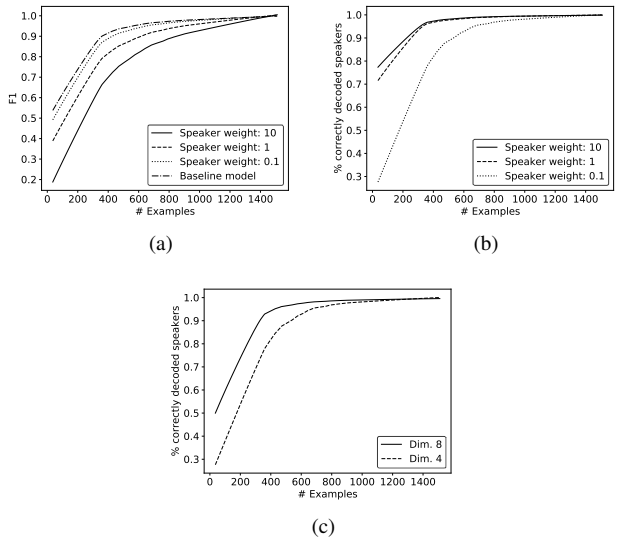


Fig. 2: Experiments with the multitask model on GRABO for different parameters, showing the effect of the speaker weight on the F1 score in (a) and on the speaker identification capability in (b), and the effect of the output dimension on speaker identification in (c).

Afterwards the multitask model with speaker identification was analysed. Multiple speaker independent experiments have been executed on GRABO, first with different factors for the speaker weight λ_s (and a fixed output dimension of 4) and second with different output dimensions. Fig. 2a shows the effect of the added speaker loss term on the F1 score, compared to the performance of the baseline model without speaker identification. Fig. 2b compares the speaker recognition of experiments with speaker weights 10, 1 and 0.1. Fig. 2c shows the result of experiments with output dimension 4 and 8 and a speaker weight of 0.1. The F1 score was the same for both experiments and is thus not shown.

Fig. 3 shows the results of cross-validation experiments performed on the Fluent Speech Commands dataset, comparing the multitask model to the baseline in a speaker independent setting. The speaker weight regularisation parameter of the multitask model has been set to 1 and the output dimension of the capsules to 16.

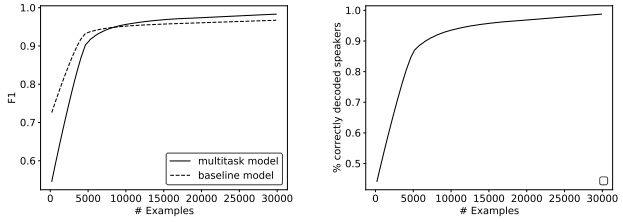
Finally the train and test experiments of [16] for the Fluent Speech Commands dataset have been replicated for comparison. Using the accuracy metric as defined in that paper, the multitask model achieved an accuracy of 97.8% on the test set after training on the partial dataset and 98.1% after training on the full dataset. These results should be compared to the model without pre-training of [16], which reaches an accuracy of 88.9% with the partial dataset and 96.6% with the full dataset.

5. DISCUSSION

From the first analysis we conclude that the output capsule dimension can be reduced without introducing errors to lower the number of parameters. The network almost solely uses the length of the vectors. If there is information in the orientation of the vector, it can be presented in two dimensions, so there is probably not much structure present in the capsules.

Fig. 2b confirms that the speakers are successfully identified in the multitask model. The performance is already at 99% after a few hundred examples, which means this task is not so difficult for the model, since there are only 11 speakers in the GRABO dataset. There is a trade-off between the speaker identification and the task learning speed, depending on the speaker weight λ_s, as presented on Fig. 2a. This negative effect on the F1 score can be explained by the fact that the task of extracting the labels is quite easy for the model on the GRABO dataset, due to the little variability in phrasings (the F1 score reaches over 90% after a few hundred examples). However based on the comparison between different dimensions of Fig. 2c, we see that now the output dimension does have an influence and the orientation of the output vector has received more meaning compared to the baseline model.

Fig. 3 shows that on the larger, more difficult Fluent Speech Commands dataset, multitask learning has improved the asymptotical performance. The learning speed is slower in the multitask model (the performance is worse when little training data is available), because the added term in the loss function will make the model initially less focused on the decoding task. Once the speakers are reliably recognised, this will help the intent decoding. With nearly 100 speakers in the dataset, the model needs enough examples to be able to make a distinction between all those speakers to identify the right one. Finally the accuracy results of the multitask model on the train and test experiment in [16] are higher than the results of the model (without pre-training) proposed there.



(a) (b)

Fig. 3: Experiments with the multitask model on the Fluent Speech Commands dataset, comparing the F1 score to the baseline model in (a) and showing the speaker recognition performance for a speaker weight of 1 in (b).

6. CONCLUSION

In this paper we investigated the use of capsule networks as fast learning models for speech-to-intent systems, or more specifically for command-and-control applications. Analysis of the basic capsule network showed that there was not much information encapsulated in the orientation of the output vector. The length of the vector is most important for the classification task.

The baseline model has been expanded to incorporate multitask learning in the capsule vectors and in order to create more structure in its output. We moved to a speaker independent setting, and the model now also learns to identify the speaker of the utterance. For this auxiliary task a linear mapping is introduced on the average output capsules to combine their dimensions and use them for learning. From the results we can conclude that this regularisation has led to structure in the output capsule, reflecting speaker identity. Furthermore identifying the speaker and encoding the required information structurally into the orientation of the capsule vectors has improved the performance of the model when the dataset is challenging and large enough. It is remarkable to see that our model performs well even on the Fluent Speech Commands dataset, where there are some speakers with only very few recorded utterances.

7. REFERENCES

[1] Heidi Christensen, Stuart P. Cunningham, Charles Fox, Phil D. Green, and Thomas Hain, "A comparative study of adaptive, automatic recognition of disordered speech," in *Interspeech*, 2012.

[2] Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio, "Towards end-to-end spoken language understanding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5754–5758.

[3] Y. Chen, R. Price, and S. Bangalore, "Spoken language understanding without speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6189–6193.

[4] Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 720–726.

[5] Vincent Renkens, *ASSIST: Assistive Speech Interface for Smart Technologies*, Ph.D. thesis, KU Leuven, 2019, `https://lirias.kuleuven.be/retrieve/531418`.

[6] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton, "Dynamic routing between capsules," in *Proceedings NIPS*, 2017.

[7] Bart Ons, Jort F. Gemmeke, and Hugo Van hamme, "Fast vocabulary acquisition in an nmf-based self-learning vocal user interface," *Computer Speech & Language*, vol. 28, no. 4, pp. 997 – 1017, 2014.

[8] Bart Ons, Jort F. Gemmeke, and Hugo Van hamme, "The self-taught vocal interface," *EURASIP Journal on Audio, Speech, and Music Processing*, p. 43, Dec 2014.

[9] Vincent Renkens and Hugo van Hamme, "Capsule networks for low resource spoken language understanding," *Interspeech*, Sep 2018.

[10] Rich Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul 1997.

[11] Natalia Tomashenko, Antoine Caubrière, and Yannick Estève, "Investigating adaptation and transfer learning for end-to-end spoken language understanding from speech," in *Interspeech*, 2019, pp. 824–828.

[12] Zhiyuan Tang, Lantian Li, and Dong Wang, "Multi-task recurrent model for speech and speaker recognition," *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1–4, 2016.

[13] "GRABO dataset," `https://www.esat.kuleuven.be/psi/spraak/downloads/`.

[14] Vincent Renkens, Steven Janssens, Bart Ons, Jort Gemmeke, and Hugo Van hamme, "Acquisition of ordinal words using weakly supervised nmf," in *Proceedings Spoken Language Technology Workshop (SLT)*. 2014, pp. 30–35, IEEE.

[15] "Fluent Speech Commands dataset," `https://www.fluent.ai/research/fluent-speech-commands/`.

[16] Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio, "Speech model pre-training for end-to-end spoken language understanding," in *Interspeech*, 2019.

[17] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto, *Modern Information Retrieval*, chapter 4, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.