Matching Network for American Sign Language Recognition

Ming Lei
minglei@cmail.carleton.ca
Department of Computer Science
Carleton University
Ottawa, ON K1S 5B6

Abstract

American Sign Language is widely used in deaf community. Many existing deep learning methods are able to perform classification task on ASL letter well. However, they are some limitation of using the neural networks such as requiring large image samples, taking time on training, etc. In this project, We employed the Matching Networks model to complete a one-shot learning task on ASL.

1 Introduction

Sign language is a natural language that used by people with impaired hearing and speech. It is a organized visual language that using facial expression, movements as well as gestures to communicate. American Sign Language (ASL) is widely used in North America [1]. Recent years, Sign Language Recognition has been a common topic in machine learning and computer vision fields. Some neural networks have been applied in ASL letter translation with accuracies over 90% [2] [3], many of them require a large data set or complicated steps in data preparation phase such as image background subtraction, contrast adjustment and high resolution images.

Due to the existing SLR approaches typically require a large number of annotated examples for each ASL letter class, we explored the one-shot learning model, specifically the Matching Networks to classify the ASL letters. In our project, we will use very limited data samples retrieved from Kaggle website [4]. To the end, we introduce the problem of one-shot American Sign Language Recognition.

2 Related Work

2.1 Neural Networks

Some neural networks have been used to classify ASL images. The advantage of using neural networks is that they learn the most important classification features. However, they require more time and many weight update using stochastic gradient descent when it comes with a large amount samples.

2.2 K-Nearest Neighbor

The non-parametric models such as K-Nearest Neighbor (KNN) doesn't require any training. For example, given two image embedding, KNN calculates the approximate distances between them and then assign the unknown image to the class of its K-nearest neighbors. However, this method makes test phase slower since the model structure is determined from the data it self and it's performance depends on the chosen metric, e.g., the similarity function to calculate the distance between data points. We will further explore the chosen metrics for our model.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

2.3 Few-shot Learning Concept

There is a gap between machine and human in the task of image classification. For example, a child can easily recognizes an apple even if he or she has seen one single apple image. In this way, inspired by the human, a few-shot learning model aims to close this gap between machine and human. Unlike the traditional supervised learning approaches that using same training and test classes, we employ the few-shot model to recognize sign letter classes that are never seen during training and It is supposed to learn only few samples in training phase. Compare to the supervised learning, the few-shot learning model separates data samples into the support set and query set. Specifically, the support set includes training data with labeled samples and the query data set includes the unlabeled samples that are be predicted by our model.

2.4 Matching Networks

In our project, we employ the Matching Networks [5] to complete the ASL Recognition task. It is a few-shot learning model that takes advantages from both parametric models and non-parametric models.

In the Matching Networks, a fully differentiable nearest neighbors classifier is trained. It first embeds all samples with a simple neural network, takes the original embedding as input to calculates full context embedding (FCE) with an bi-LSTM [6] model on support set and query samples, then calculates the cosine distance between the embeddings of query samples and support sets to make a class prediction by taking the weighted average of the support sets with normalized distance using an attention kernel.

3 Experiment

3.1 Data Set

The ASL Alphabet Test data set [4] has been used to train the Matching Network model. The data set is not large, it contains 870 images, each image contains a had making the shape of an ASL letter. There are 29 classes including letter A-Z delete, space and nothing in total. Each class contains 30 200x200 pixel images.

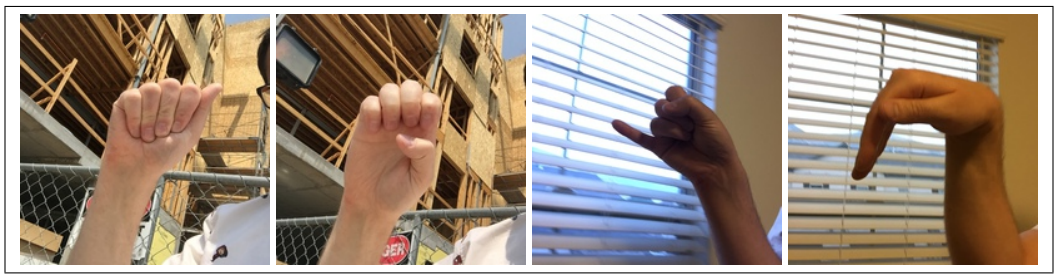


Figure 1: Samples for ASL sign A, E, J and Delete.

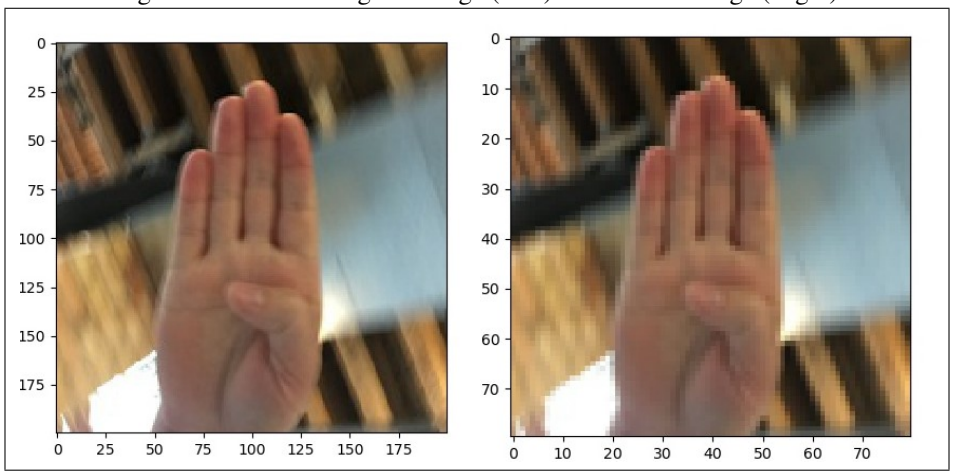
Table 1: Test Accuracy with Image Rotation & No Rotation

Dataset	No Rotation	Rotation
Omniglot	93.75%	98.28%
American Sign Language	50.39%	69.53%

3.2 Data Preparation

The data Preparation was done using the *PILLOW* library and an *imageio* library, they are used to load images and convert to *numpy* data format *.npy* file. To speed up the learning and save the time, we also reduced image size from 200x200 pixels to 80x80 pixels. See Figure 2 as an example. We follow the procedure of Vinyals et al. [5] by augmenting the character classes with rotations in multiples of 90 degrees, since it would increase accuracy of machine learning models. See table 1.

Figure 2: Letter B: Original Image (Left) vs. Resized Image (Right)



3.3 Training Strategies

Let denote the predicted label as $\hat{y}$
Lets consider a dataset with set of D belongs to L classes. We define a support set S as in Equation 1.

$$Eq.1 : S = \{(x, y)|x \in D, y \in L\}$$

The training set $\{(t_1, s_1), (t_2, s_2), ..., (t_N, s_N)\}$ has N elements, where each target instance, t_i is a training instance pair, (x, y), and is never featured in its' support set. The Matching Networks model θ, learns to recognise the class, y, for a given target image, x, relative to support set, S. Thus, we can predict the label, $\hat{y}$, for a given image, $\hat{x}$, relative to a support set, S.

$$Eq.2 : \theta(\hat{x}, \hat{S}) \rightarrow \hat{y}$$

where $\hat{S}$ is the embedding of images in support set S Since the support set doesn't not include the class $\hat{y}$, which $\hat{x}$ belongs to, the results may not outperform to supervised learning models.

For each support set S, we choose, 5 different classes and 1 image for each class, this is called 5-way 1 shot Test. Similarly, we also test 5-way 5-shot for all data sets we selected.

In the experiment, the following parameters are used:

- batch_size: 16
- total_epochs: 500
- patience: 20 (for early stop)
- total_train_batches: 100 (total number of batches for training)
- total_validation_batches: 32 (total number of batches for validation)
- total_validation_batches: 32 (total number of batches for test)

3.4 Experiment Design

The cross-entropy function is used to determine the loss and is given by:

$$L(\theta) = -\frac{1}{n}\sum_{i}^{n}[\, y_i \log(p_i) \; + (1 - y_i)\log(1 \; - p_i)] = -\frac{1}{n}\sum_{i}^{n}\sum_{j}^{m} y_{ij}\log(p_{ij})$$

In this project, I attempted to using following methods to compute the similarity between images in support set and the query image:

The cosine distance:

$$cosine\ distance = \frac{x_1 \cdot x_2}{max(||x_1||_2 \cdot ||x_2||_2, \epsilon)}$$

The pairwise distance:

$$||x||_p = (\sum_{i=1}^{n} |x_i)^p)^{1/p}$$

The euclidean distance:

$$euclidean\ distance = \sqrt{(p-q)^2}$$

The CNN configuration that used to embed support set and query set:

Table 2: Accuracies on Data sets

Dataset	5-way 1-shot	5-way 5-shot
Omniglot (Paper)	98.1%	98.9%
Omniglot (Our)	95.12%	98.63%
American Sign Language	69.53%	74.22%

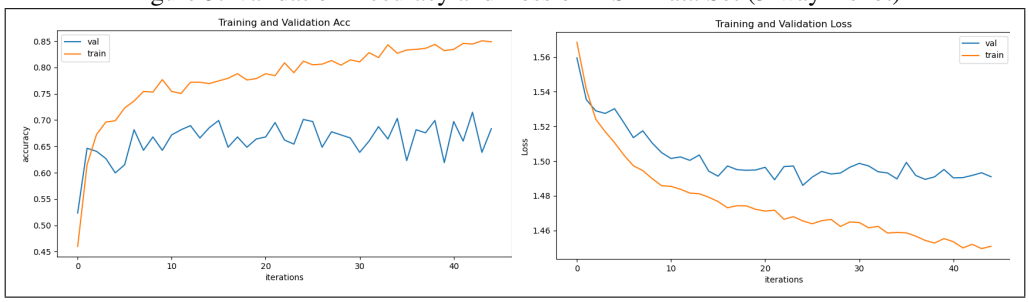
Table 3: Accuracies with Different Similarity Functions (5-way 1-shot)

Dataset	Cosine	Pairwise	Euclidean
American Sign Language	69.53%	22.07%	50.3%

4 Results

4.1 Test k-way n-shot

Figure 3: Validation Accuracy and Loss on ASL Data Set (5-way 1 shot)

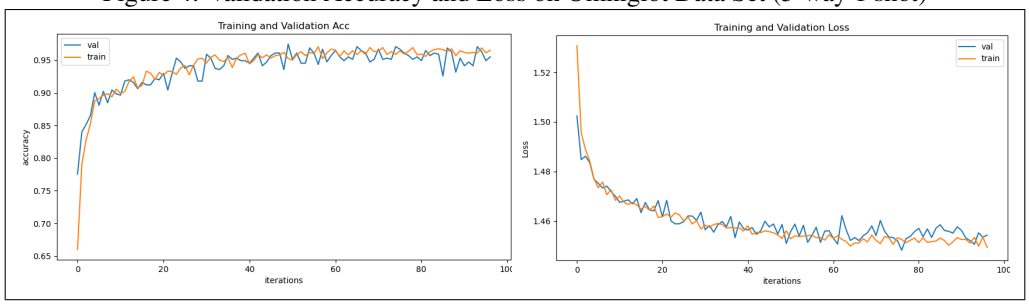


According to the 5-way 1-shot table (see Table 2), we can find that all 5-way 5-shot results are outperform to 5-way 1-shot results since 5-shot provide more images during training. In this project, result of Omniglot data set is used to verify my implementation on Matching Network and to compare the results from the paper. The results on Omniglot data set is near to 96% which is close to the result on Matching Networks paper. There is not a surprising outcome since Omniglot are single channel images and they are simple character.

The results of the American Sign Language experiments are shown in Table 2 as well. However, the classification on the American Sign Language is more challenge than Omniglot. The 5-way 1-shot and 5-way 5-shot are 69.42% and 74.22% respectively. Although, it doesn't outperform to some deep learning methods that are mentioned in section 1, it is still an acceptable result. The ASL data set has only 30 samples for each class and most samples come with a complex background. Compare with other methods, the one-shot learning Matching Network doesn't require large samples and background removal.

Lastly, I attempted to apply different similarity functions but the their performance are not good, see table 3.

Figure 4: Validation Accuracy and Loss on Omniglot Data Set (5-way 1 shot)



5 Conclusion

In this project, the Matching Network model is reproduced and used to complete a one-shot learning task on American Sign Language model and the accuracy reaches around 70%. Additionally, I further explored the similarity metrics that used in the Matching Network model.

References

[1] Mitchell, Ross; Young, Travas; Bachleda, Bellamie; Karchmer, Michael (2006). "How Many People Use ASL in the United States?: Why Estimates Need Updating" (PDF). Sign Language Studies (Gallaudet University Press.) 6 (3). ISSN 0302-1475.

[2] P. Mekala et al. Real-time Sign Language Recognition based on Neural Network Architecture. System Theory (SSST), 2011 IEEE 43rd Southeastern Symposium 14-16 March 2011

[3] Y.F. Admasu, and K. Raimond, Ethiopian Sign Language Recognition Using Artificial Neural Network. 10th International Conference on Intelligent Systems Design and Applications, 2010. 995-1000.

[4] Dan Rasband. ASL Alphabet Test. url: `https://www.kaggle.com/danrasband/asl-alphabet-test` (accessed: 12.12.2021).

[5] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., "Matching networks for one shot learning," in Advances in neural information processing systems, pp. 3630–3638, 2016

[6] S Hochreiter and J Schmidhuber. Long short-term memory. Neural computation, 1997.