International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

Sign Language Translation Using Ladder Network

Chandramani

B.E from Netaji Subhas Institute of Technology, New Delhi, India

Abstract: ***In the realm of multi-modal communication, sign language is, and continues to be, one of the most understudied areas. In line with recent advances in the field of deep learning, there are far reaching implications and applications that neural networks can have for sign language interpretation In this paper translation of sign language network into text is being done using significantly fewer lower labelled data than the traditional model. By using convolution Ladder Network ,it us demonstrated how it can perform this task by achieving superior performance while using 80% less data as compared to the traditional convolution neural network.***

1. Introduction

Sign Language is a unique type of communication that often goes understudied.

While the translation process between signs and a spoken or written language is for-mally called 'interpretation,' the function that interpreting plays is the same as that of translation for a spoken language.In our research, we look at American Sign Language (ASL), which is used in the USA and in English-speaking Canada and has many different dialects.

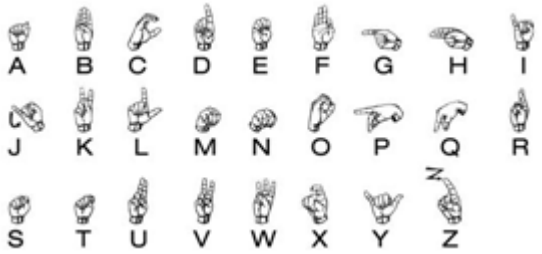


Figure 1: Sign Language Alphabet

spelling is used. Finger spelling is a method of spelling words using only hand gestures. One of the reasons the finger spelling alphabet plays such a vital role in sign language is that signers used it to spell out names of anything for which there is not a sign. There is a lot of unlabeled sign language data available (For e.g. recording sign language conversations; video of sign language interpreters at public) , but it is time consuming events to label this data because each word in the English dictionary requires a separate label.

A ladder network can greatly reduce the amount of labels required and make this task more feasible. This research paper aims to ex-plains ladder network which is a special type of neural network and how it can be used to acheive superior accuracy with less labeled data than the traditional CNN model. The input for this model will be the images of the hand signs and the output will be a one-hot encoded vector specifying the translated word.

There are 22 hand shapes that correspond to the 26 letters of the alphabet, and you can sign the 10 digits on one hand. One of the nuances in sign language is how often finger

2. Related Work

In recent years Convolution Neural Networks (CNN) have been extremely successful in im-age recognition and classification problems, and have been successfully implemented for human gesture recognition. In the realm of Sign Language Recognition there has been proper work done using deep CNN's, with input- recognition that is sensitive to more than just pixels of the image.

There have been prior work on translation of sign language images to text. Vivek Bheda and N.Dianna Radpour used CNN's to convert sign images to text [1]. They used a data set with about 67 images per class [5] to achieve 82.5% accuracy [1]. In this research paper we aim to improve on both accuracy and reduced number of labels since both human and Bayes error are close to 100% accuracy.

There has been prior work on using lad-der networks for classification using minimal labeled data. Ladder Networks have been used for classifying handwritten digits which achieved about 98% accuracy and it used only 10 labels per class [3]. Ladder Networks have been used in Human Activity classification [4] and sequence models [6].

3. Data

For this research purpose, Sign Language Dataset from Kaggle [**?**] has been used as an experimental data set. This datset contains all the alphabets of the English Language except J and Z because they are signed using motions which is out of scope for this paper but in fu-ture the principle established here can be used in motion capturing RNN models. This dataset contains total of 23 classes.

This dataset contains 27455 training images, 3586 validation images and 3586 test images. Each image is 28x28 pixels and is already grey-scaled. Each pixel from the training images became an input feature. The images were normalized so that pixel values to be between 0 and 1. The images were not augmented as the ultimate goal of this research is to use fewer labeled images.

There is an average of about 1144 images per class and is evenly distributed. The standard deviation of each class is about 81.96 and a max/min of 1294 and 957 respectively. The data- set contains labeled data but it is ignored as goal is to use as much less labeled data as possible.

Figure 2: Sample Images from the Dataset

4. Methods

4.1 The Ladder Network Architecture

The Ladder Network Architecture is de-scribed in this section. Consider a data set with N labeled examples (x(1),y*(1)), (x(2),y*(2)),..... (x(N),y*(N)) and M unlabeled examples x(N+1),x(N+2),..... x(N+M) where M
N. The objective is to learn a function that models $P_{(y|x)}$ by using both labeled example and the large quantity of the unlabeled exam-

ples. Ladder Network contains this function called deep Denoising Auto Encoder (DAE) in which noise is injected into all hidden layers and the objective function is weighted as the sum of the supervised Cross Entropy cost on the encoder and the unsupervised denoising Square error cost at each layer of the decoder. Due to addition of noise in all hidden layers get corrupted, another encoder with shared parameter is responsible for providing the clean reconstruction targets i.e the noiseless hidden activation layers. (See Figure 3).

Explicitly, this is how Ladder Network is

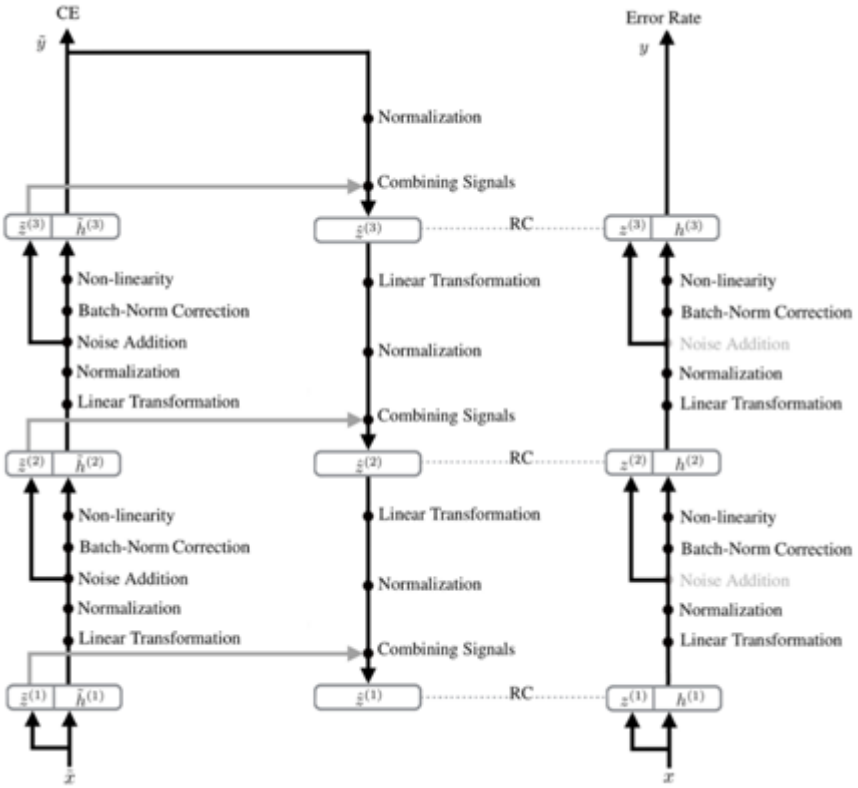


Figure 3: As shown , there are two encoders on each side and one decoder in the middle of Ladder Network. $z^{(l)}$ and $z\tilde{}^{(l)}$ are computed at each layer by linear optimization and normalization on h-(l 1)-˜ (l 1)-. h-(l 1)-˜ (l 1)---and h---and h-

are obtained by applying Batch Normalization correction and non-linearity. By combining two streams of information , the lateral connection represented by gray line $z\tilde{}^{(l)}$ and vertical connection $u^{(l+1)}$, $z\hat{}^{(l)}$ is constructed. Here, CE stands for Cross Entropy and RC stands for Reconstruction Cross Entropy. The final objective function is the weighted sum of all Reconstruction cost and Cross Entropy cost. [2]

Defined [2]:

$$x\tilde{}, y\tilde{}, z\tilde{}^{(1)}, ..., z\tilde{}^{(L)} = \textbf{Encoder}_{noisy}(x) \quad (1)$$

$$x\tilde{}, y\tilde{}, z\tilde{}^{(1)}, ..., z\tilde{}^{(L)} = \textbf{Encoder}_{clean}(x) \quad (2)$$

$$x\tilde{}, y\tilde{}, z\tilde{}^{(1)}, ..., z\tilde{}^{(L)} = \textbf{Decoder}(z\tilde{}^{(1)}, ..., z\tilde{}^{(L)}) \quad (3)$$

Here x is input, y is noiseless output and y˜ is noisy output. $z^{(l)}$ is the hidden representation, $z\tilde{}^{(l)}$ is the noisy version , and $z\hat{}^{(l)}$ is the reconstructed version at any given layer l .In this case Encoder and Decoder is replaced by multi-layer perceptron but it can be replaced by any multi-layer

architecture. The objective function is weighted sum of unsupervised costs and supervised cost.

4.2 Cost Function

The loss function is the combination of the clas-sification loss i.e cross entropy loss from the encoding DNN and the sum of all the layer's reconstruction cost which is the difference be-tween de-noised value ($z\hat{}^{(l)}$) and clean z value ($z\hat{}^{(l)}$) . For Convolution ladder Networks, only the last layer's reconstruction cost is used.

$$\text{Cost} = \sum_{n=1}^{N} \log P(\tilde{y}(n) = y(n) | x(n)) + \sum_{n=N+1}^{M} \sum_{l=1}^{L} \lambda_l \, \text{ReconsCost}(z^{(l)}(n), \hat{z}^{(l)}(n))$$

The reconstruction cost forces the network to learn a set of weights that result in similar entities also having similar z values. The loss will be high if slight changes in one layer's z value causes cast difference in other layers z values. Therefore, similar entities have same z values causing similar prediction from the DNN. Subsequently, data that hasn't been seen by the model will be accurately classified be-cause their z values will be similar to that of the labeled data, and they will be classified accordingly. By combining the cross entropy loss and the reconstruction loss , the ladder network learns weight distributions such that it generates accurate predictions and can also generalize the unseen data.

Here for this classification purpose, convo-lution layers has been used within the ladder network as they are more suitable for image classification and they allow a filter which can be used in multiple locations on the image and are less susceptible to noise. Convolution Ladder Network are similar to regular Ladder Networks except their reconstruction cost is only comprised of the last layer's reconstruc-tion cost.[1]

5. Experiment

For this experiment baseline CNN model and ladder network has been trained using differ-ent numbers of labeled example. Results from these models were used to demonstrate how the ladder network can achieve superior accu-racy with fewer labeled examples.

Primary metric for this purpose is accuracy because it is the best measure of whether we are translating sign language correctly, or not. One of the advantage of using this data set is classes are roughly balanced so we don't have to worry about imbalanced classes inflating the accuracy.

5.1 Model Architecture

Convolution Ladder Network used has the following architecture:

- 32 3x3 filters
- 64 3x3 filters
- 128 3x3 filters
- 128 fully connected layers and a
- Softmax layer

Finding the optimal architecture was an iter-ative process. It was done by using following steps:

1) The experiment was started with a fully connected network without any convolu-tion layer but the variance was too high. This indicated that this model was over-fitting. Using convolution layers reduced over-fitting because they are less suscepti-ble to minor feature skews and offsets.
2) When first tried using convolution layer ,it faced an under-fitting problem as ev-idenced by a high avoidable bias. The network originally used 16,32 and 64 3x3 filters in a 3 layer configuration. To avoid this problem more filters were used and fully connected layers was changed into two layers (500,250).
3) The previous step again led to over-fitting. This was solved by changing the fully con-nected layers to a single layer with 128 nodes.

5.2 Hyperparameter Tuning

This step was to tune hyperparameters for optimal model. For this, coarse-to-fine process is used. For this network, following were found optimal:

- Batch Size : 32
- Learning rate : 0.001
- Denoising cost of last layer (multiplied with reconstruction cost) : 3
- Noise standard Deviation : 3

Table 1: Result Table.

Model Architecture	Number of Labeled Data Per Class	Accuracy	Precision	Recall
Baseline CNN	20	0.53	0.55	0.511
Ladder Network	20	0.66	0.64	0.648
Baseline CNN	40	0.66	0.64	0.65
Ladder Network	40	0.84	0.79	0.80
Baseline CNN	80	0.78	0.77	0.77
Ladder Network	80	0.89	0.83	0.844
Baseline CNN	100	0.82	0.80	0.80
Ladder Network	100	0.92	0.88	0.89
Baseline CNN	200	0.83	0.81	0.82
Ladder Network	200	0.96	0.90	0.92

6. Result and Error Analysis

6.1 Result

These results were evaluated using the test data set (See Table 1).

The convolution Ladder Network clearly out-performed the baseline CNN models in all the situations.This is expected,given the ladder net-work includes a reconstruction cost that allows it to better generalize all data. The Ladder Net-work required only 40 labels to achieve higher accuracy while CNN model used 200 labels to reach that accuracy which is reduction of approximately 80 % of data.

6.2 Error Analysis

The given model is over-fitting which is ex-pected because deliberately because as possible very little data is used. As such, the model doesn't have enough data to generalize to un-seen cases very well. In all the above cases given in the table training accuracy was about 98%. As the difference between between train-ing and validation error is many times the dif-ference between training and human error, the variance is very high for most cases except for the model trained on 200 labeled examples. This combined with the high training accuracy indicates overfitting.

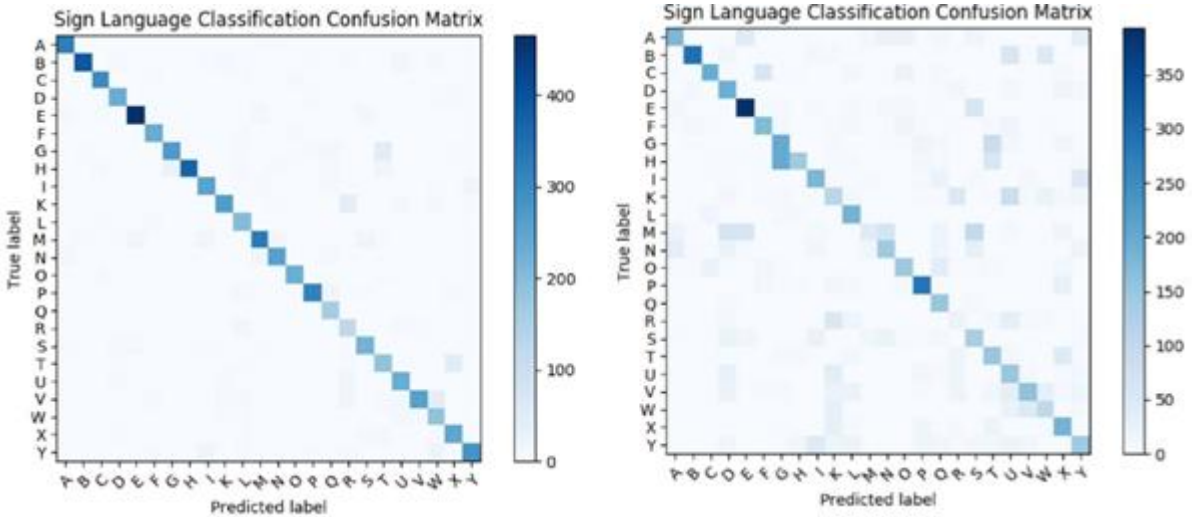
There is also some avoidable bias. Since sign language recognition is easily performed by humans with a higher degree of accuracy therefore human error and Bayes error should be very similar to each other and both should be close to 0. But since Ladder Network add Gaus-sian noise into each layer the small amount of avoidable bias is expected which is about 2 % since training accuracy is about 98 %.

6.2.1 Confusion Matrices

The confusion Matrices (See figure 4) indicates which classes the model is likely to mis-predict, and what the mis-predicted class is. The first matrix is for the model trained using 200 la-beled example and the second is the one for the model trained with 10 labeled example.

Figure 4b. has a higher rate of mispredic-tions because the model uses less data and therefore generalizes much less.

Using the confusion matrix we can see some of the most common classes of the mispredic-tions. G is often misclassified as T, which isn't surprising since they both involve a protruding finger from a fist. It's also often misclassified as an H , since they are almost identical (except for one finger.) T and H are different enough from each other such that we don't see many misclassifications between them. (See figure 5). As expected signs that look very similar are more likely to be misclassified with each other because the model doesn't have enough data to learn the details that can generalize for distinguishing between the two classes.



(a) 200 labels per class (b) 10 labels per class

Figure 4: Confusion Matrices

(a) Sign of G (b) Sign of H (c) Sign of T

Figure 5: Sign Language Alphabet which is mispredicted

7. Conclusion and Future Work

In this research paper it is demonstrated how a convolution ladder Network can achieve 25% higher accuracy than traditional convolution neural networks in situations when there isn't a lot of labeled data. The ladder network required approx 80% less data to exceed the performance of a baseline CNN. This greatly reduces the amount of labels required.

This concept can be extended to other im-age recognition problems as well ,where it is difficult to obtain a labelled data. In model training reducing the amount of labelled data can be of great assistance.

This ladder network can be expanded using number of words in the sign language translation model supports . It can be also explored RNN ladder networks for translating sign langauage videos. This system can be also product-

ionize by build-ing in image localization to identify signs in a larger image and classify them in real time.

References

[1] Vivek Bheda and N. Dianna Radpour.. Using Deep Convolutional Networks for Gesture Recognition in American Sign Language. arXiv preprint arXiv:1710.06836,2017.
[2] Mohammad Pezeshk, Linxi Fan, Philemon Brakel, Aaron Courville, Yoshua Bengio Deconstructing the Ladder Network Architecture arXiv:1511.06430,2016
[3] Antii Rasmus , Hari Valpola, Mikkio Honkala , Mathias Berglund Semi Supervised Learning with Ladder Networks arXiv preprint arXiv:1507.0262, 2015
[4] Ming Zeng, Tong Yu, Xiao Wang , Le T. Nguyven, Ole J. Mengshoel, Ian Lane Semi-Supervised Convolutional Neural Networks for Human Activity Recognition arXiv preprint arXiv:1801.07827, 2017
[5] Barczak, A.L.C., Reyes,N.H, Abstillas, M. , Piccio, A, Susnjak,T. A new 2D static hand gesture colour image dataset for ASL gestures, Research Letters in the Information and Mathematical Sciences 15, 12-20 https://mro.massey.ac.nz/handle/10179/4514, 2011
[6] Isabeau Premont-Schwartz, Alexander Ilin, Tele Hotloo Hao, Anti Rasmus , Rinu Boney, Harri Valpola Recurrent Ladder Networks arXiv preprint arXiv:1707.09219, 2017
[7] Sign Language MNIST (Version 1) from Kaggle [Drop-In Replacement for MNIST for hand gesture Recognition Tasks]. 2018 https://www.kaggle.com/datamunge/sign-language-mnist