M2015: CSE577 Machine Learning
Assignment 3

Introduction to Deep Learning

Siddhant Prakash
201201008

1 Introduction

Deep Learning is the branch of machine learning which attempts to represent data at a higher level of abstraction with the help of complex multiple processing layers between the input layer and the final output layer. Today we see a multitude of problems being solved with the help of deep learning algorithms. Although the underlying mathematical model of the network is hard to understand, it is observed that the results come out surprising well.

Deep learning are mainly based on distributed representations. The underlying assumption behind distributed representations is the fact that different data are generated by different features interacting with different multiple levels. Deep learning attempts to extract these multi-level features individually in the form of different layers of abstraction and composition. It achieves so by varying the network structure of each model which provides the learner with a unique architecture every time and thus different level of abstractions.

In this assignment, we explore Convolutional Neural Networks, its architecture and how it is applied with hands-on-practice with the CIFAR-10 and CIFAR-100 datasets. Section 2 of the report discusses a recent technical research paper, which uses deep learning algorithms to achieve a novel goal of daily activity recognition using egocentric images. Section 3 and section 4 gives us the opportunity to explore various parameters on which CNNs depends by designing experiments on CIFAR-10 and CIFAR-100 dataset. In Section 5 we revisit DropConnect which is a generalization of the classic DropOut algorithm. Finally, we summarize in section 6 where we provide link to the supplementary codes wrote for the assignment.

2 Discussion on one recent success of deep learning

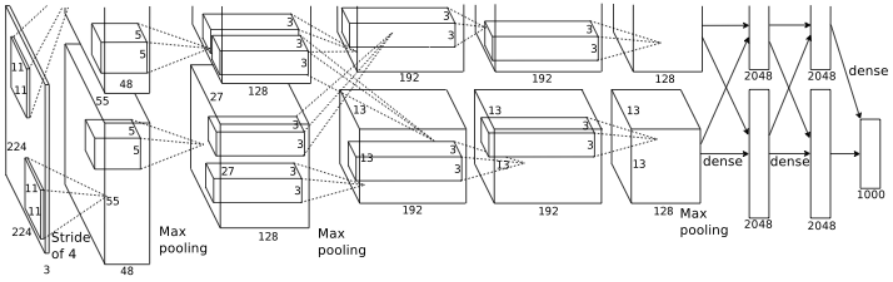


Fig. 1: The CNN network architecture used to train the classifier

The paper I would like to discuss is "Predicting Daily Activities From Egocentric Images Using Deep Learning."[1] presented at The 19th International Symposium on Wearable Computers, September 2015.

The work uses Convolutional Neural Networks with a novel classification method introduced in the paper as late fusion ensemble. This late fusion ensemble incorporates relevant contextual information such as day of the week and time, which increases the classification accuracy and provides state-of-the-art results on the data collected over commonly used methods, such as a traditional CNN or a Classic Ensemble. The work further determines the amount of data that is needed to train an initial CNN classifier for daily activity recognition and amount of data required to fine-tune the model on a per-user basis.

The network has five convolutional layers, some max-pooling layers, and three fully-connected layers followed by a dropout regularization and a softmax layerwith an image size of 256X256, just as in Figure 1 from [3]. The dataset consisted of 40,103 egocentric images, collected over a 6 month period with 19 activity classes. The given dataset was divided into 75% training, 5% validation and 20% test sets. The parameters were set as base learning rate to be 0.0001 with the same momentum of 0.9 and weight decay of 0.0005 again similar to [3].

One of the assumptions made in the paper is that the images will not have any class overlap. For example, the learned network classified worst with the classes labelled as "Chores" or "Chatting" which the classifier confuses with "Cleaning", "Working" and "Family". The latter could be attributed to the reasoning that when the subject is conducting a chore, the family is in background or chores could easily come under working in a broader sense.

The baseline methods used are (i) kNN classier trained on metadata and the color histogram gives an accuracy of 73.07% and (ii) Random Decision Forest(RDF) classifier with 500 trees trained on metadata and color histogram with a slightly better accuracy of 76.07%. The only plausible reason of the network performing better is that the network is able to better relate the local features learned through images with the metadata such as time and day to create an overall routine of the task. The same could be understood through Figure 2.

The possible drawbacks are the class overlap in the dataset. It can be further improved by making the dataset more specific, possibly with large number of labels and removing the class overlap. Another method could be using videos as dataset, but the problem lies with capturing these kind of videos with limitation in requirements of tethered power and bandwidth. Maybe small videos captures, like vines[2] could be a possible solution to the

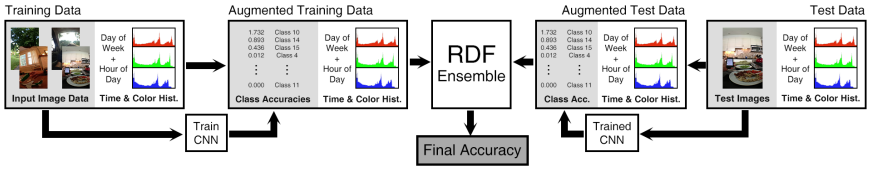


Fig. 2: Overview of the CNN Late Fusion Ensemble for predicting daily activities

issue. Another improvement mentioned in the paper is data augmentation while training to prevent over-fitting and increasing accuracy.

3 Experiment 1: Convolutional Neural Networks using CIFAR-10 dataset

In this section we learn to independently train a convolutional neural network and observe the effectiveness of the representation learnt using raw pixels as compared to that learnt by this network.

3.1 Dataset

CIFAR-10 dataset: The dataset is divided into five hundred training batches and one hundred test batch, each with 100 images. There are 10 mutually exclusive classes with 6000 images per class.

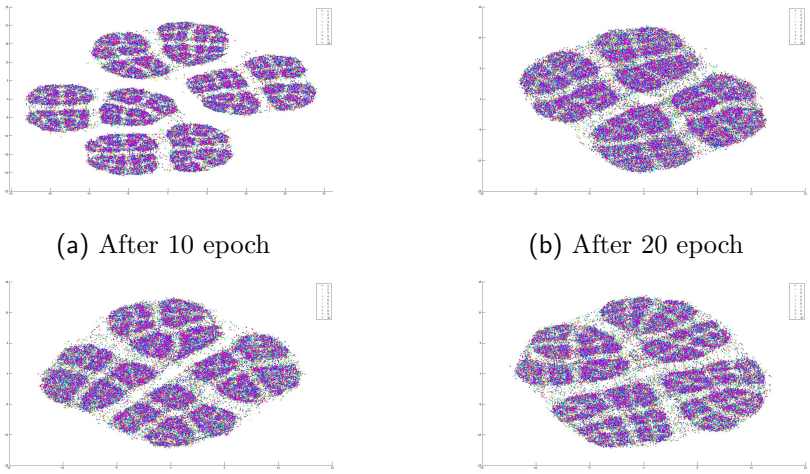
We perform two experiments on the CIFAR-10 dataset. The network structure and results are reported individually as follows:

We use Matconvnet library for our implementation. The network structure consist of a total of 19 layers with 5 convolutional layer, 3 pooling layer, 4 bi-linear normalization layer and few dropout layers. We use a softmax classifier on top over the last fully connected layer for classification. The base learning rate was set to 0.01 with momentum of 0.9 and weight decay of 0.005.

e_1 : We train a CNN using raw pixel as features with a softmax classifier on top of it for 100 epochs.

The overall accuracy of the softmax classifier at the end of 100 epoch came out to be **80.45%** as learnt by the network. The confusion matrix for the softmax classifier for the inter-class labels are shown in Figure 4a

The separability of classes after every 10-epochs can be seen in Figure 3 , 5 and 6



(a) After 10 epoch (b) After 20 epoch

(c) After 30 epoch (d) After 40 epoch

Fig. 3: Separablility of classes improving with progress in training(1)

e_2 : The trained model from e_1 was used, and after removing the top-most softmax layer, the feature vectors were recomputed and passed to a train a SVM.

The overall accuracy of the SVM classifier came out to be **9.8%** as learnt by the network. The confusion matrix for the SVM classifier for the inter-class labels are shown in Figure 4b

Thus, we can infer that training a SVM using pre-trained CNN **does not** help as compared to learning the classifier from raw pixel.

C =

14	5	22	30	14	3	3	6	2	2
7	7	20	38	11	4	2	5	5	10
15	0	21	32	18	2	0	6	1	7
7	5	31	34	10	3	0	6	5	4
12	4	24	49	6	6	0	7	2	2
4	0	11	32	7	13	1	7	6	4
10	3	22	43	5	10	0	7	4	5
8	6	19	39	10	3	1	8	5	3
8	4	13	40	9	9	2	2	3	2
8	2	14	38	7	8	2	3	0	1

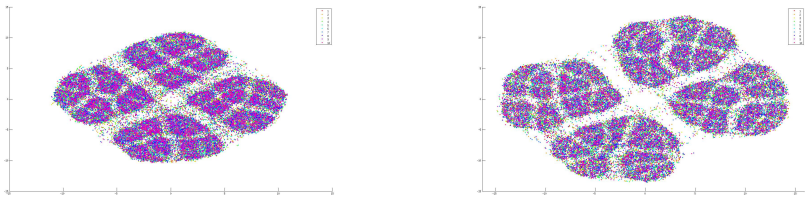
(a) For network learned using raw pixel

C =

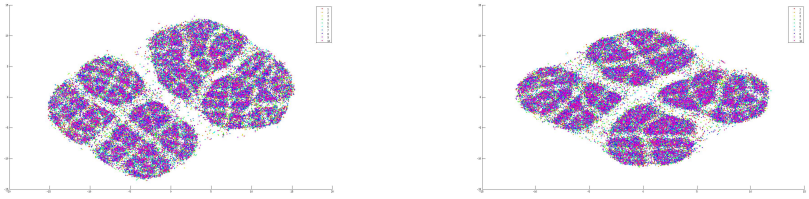
108	118	53	134	83	84	127	79	122	92
105	93	52	142	106	80	114	81	112	115
99	125	44	123	96	72	136	77	125	103
102	109	41	135	103	77	123	88	133	89
112	104	49	156	95	81	116	85	109	93
115	122	47	116	102	90	116	73	127	92
115	100	64	127	95	85	121	85	102	106
87	103	61	126	98	90	138	82	133	82
102	128	48	113	93	92	124	82	121	97
108	98	58	133	88	87	145	58	128	97

(b) For network learned using trained features

Fig. 4: Confusion matrices for classifiers learned on CIFAR-10 dataset



(a) After 50 epoch (b) After 60 epoch



(c) After 70 epoch (d) After 80 epoch

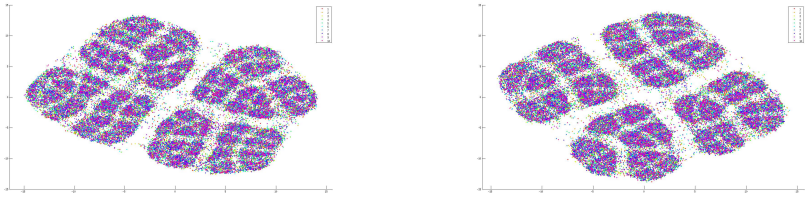
Fig. 5: Separablility of classes improving with progress in training(2)

4 Experiment 2: Parameter tuning using CIFAR-100 dataset

In this section we learn to adapt a learned representation to a similar dataset by fine tuning the parameters. We train teh CNN in fine tune mode to change the hyperparameters and get the best representation.

4.1 Dataset

CIFAR-100 dataset: The dataset is similar to the CIFAR-10 dataset. Originally, the data has has 100 classes containing 600 images each. The 100 classes in the CIFAR-100 are grouped into 20 superclasses which we use



(a) After 90 epoch (b) After 100 epoch

Fig. 6: Separablility of classes improving with progress in training(3)

as our classes, i.e. we learn the model for these 20 supreclass. Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs).

We perform two experiments on the CIFAR-100 dataset, similar to CIFAR-10. The network structure and results are reported individually as follows:

We use Matconvnet library for our implementation. The network structure consist of a total of 19 layers with 5 convolutional layer, 3 pooling layer, 4 bi-linear normalization layer and few dropout layers. We use a softmax classifier on top over the last fully connected layer for classification. The base learning rate was set to 0.01 with momentum of 0.9 and weight decay of 0.005, for e_4 and first iteration of e_3.

e_3 : Using the trained model for e_1 which had an accuracy of 80.45%, we chopped of the softmax layer and the last fully connected convolutional layer. We replace them with a new fully connected layer as per the requirements of CIFAR-100 dataset, and add a new softmax layer. The experiment is repeated three times with hyper-parameters tweaked and the results are reported.

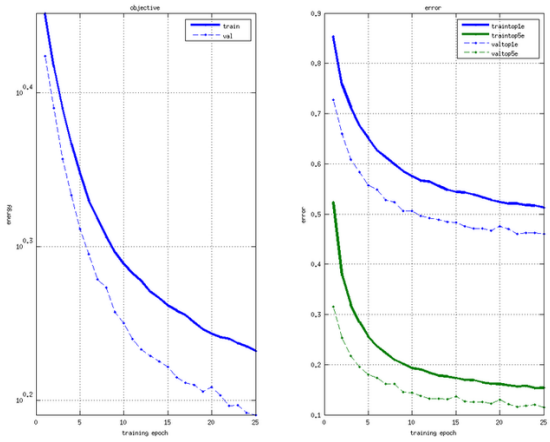


Fig. 7: Objective and Error Plot for e_3, iter. 1, learning rate 0.01

In iteration 1, The overall accuracy of the softmax classifier at the end of 25 epoch came out to be **53.93%** as learnt by the network.

The error and objective plots of the experiment is shown in Figure 7

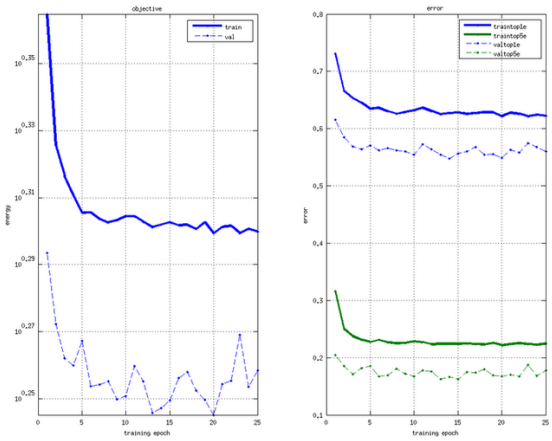


Fig. 8: Objective and Error Plot for e_3, iter. 2, learning rate 0.1

In second iteration the network structure is kept same with the learning rate set to 0.1 with momentum of 0.9 and weight decay of 0.005, increasing the learning rate. The overall accuracy of the softmax classifier at the end of 25 epoch came out to be **43.95%** as learnt by the network.

The error and objective plots of the experiment is shown in Figure 8

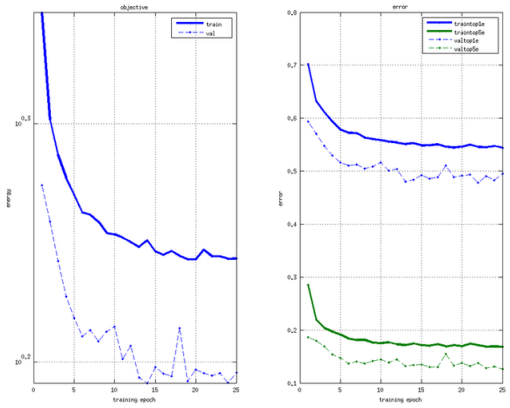


Fig. 9: Objective and Error Plot for e_3, iter. 3, learning rate 0.05

In third iteration the network structure is kept same with the learning rate set to 0.05 with momentum of 0.9 and weight decay of 0.005, increasing

the learning rate. The overall accuracy of the softmax classifier at the end of 25 epoch came out to be **50.50%** as learnt by the network.

The error and objective plots of the experiment is shown in Figure 9

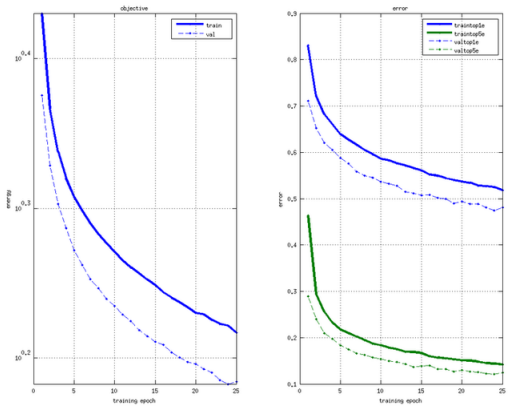


Fig. 10: Objective and Error Plot for e_3, iter. 4, momentum 0.5

In fourth iteration the network structure is kept same with the learning rate reverted back to 0.01 with momentum decreased to 0.5 and weight decay of 0.005. The overall accuracy of the softmax classifier at the end of 25 epoch came out to be **51.93%** as learnt by the network.

The error and objective plots of the experiment is shown in Figure 10

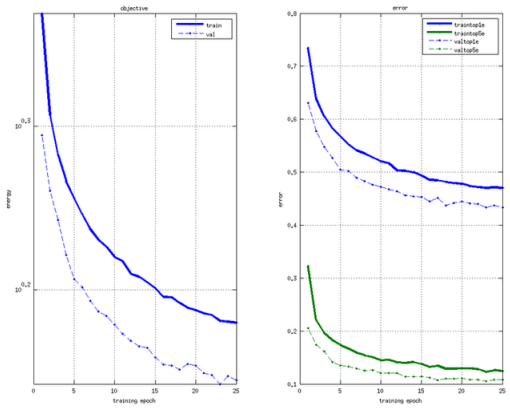


Fig. 11: Objective and Error Plot for e_4

e_4 : We train a CNN using raw pixel as features with a softmax classifier on top of it for 25 epochs.

The overall accuracy of the SVM classifier came out to be **58.47%** as learnt by the network.

The error and objective plots of the experiment is shown in Figure 11

Thus, we can see that learning a model from scratch performs better than fine-tuning a learned model from similar datasets in this case. Where the accuracy of the fine-tuned model couldn't exceed 53.93% even with tweaking the learning rate and momentum, the model learnt from scratch performed better with an accuracy of 58.47%. The reason for this observation maybe attributed to the fact that raw pixel helps in better recognition of the features by multiple intermediate "filter" layers whereas in a learned model the performance suffers due to incorrectly learned labels because of similar classes as a result of class overlap in the dataset.

Implementation Details

All the experiments were performed on either of the 2 GPUs 'Tesla C2070' or 'GeForce GTX 480' installed on the same system with GPU option set to true for matconvnet library. For e_1 it took approximately 3-4 Hrs. to train the classifier for 100 epochs. For e_2 we used libSVM classifier[6] and the model took 25-30 minutes to learn possibly because of the erroneous feature extracted. For each of the experimens performed for e_3 and e_4 it took about 15-20 mins. to train for 25 epochs.

5 Advances made in Convolutional Neural Networks and their performance on the CIFAR-10 and CIFAR-100 dataset

The paper assigned to me as result of mod(201201008,5) + 1 = 4, "Regularization of Neural Networks using DropConnect"[4], presented at the 30^{th} International Conference on Machine Learning (ICML-13). 2013.

The paper introduces a novel way of regularization of neural network with the help of DropConnect which is a generalization of the famous dropout algorithm[5]. When training with Dropout, a randomly selected subset of activations are set to zero within each layer. In DropConnect instead of the activations being set to zero, the weights within the layer are set to zero randomly. Thus each unit receives input from randomly selected input from previous layer hence changing the network architecture drastically with each iterations.The given algorithm allows us to train large model

while avoiding over-fitting which yields state-of-the-art results.

On CIFAR-10 dataset, two experiments are performed with two different network structure. Both the network uses the simple convoutional network feature extractor described in [3], Figure 1. On top of the 3-layer feature extractor, 64 fully connected layer using No-Drop, Dropout or DropConnect is added. The network is learned for 150-0-0 epochs with 0.001 as the learning rate. In the second experiment, there are 2 convolutional layer and 2 locally connected layer, along with a 128 neuron fully connected layer with relu activation, which is added between the softmax layer and feature extractor. The images are further cropped from 32X32 to 24X24 and the classifier is learned at a learning rate of 0.001.

6 Summary and comments

Video, Final Codes & Supplementary link : ML_Assignment Supplementary

References

[1] D. Castro, S. Hickson, V. Bettadapura, E. Thomaz, G. Abowd, H. Christensen, and I. Essa (2015), *"Predicting Daily Activities from Egocentric Images Using Deep Learning,"* in Proceedings of International Symposium on Wearable Computers (ISWC), 2015.

[2] The Vine Network `https://vine.co/`

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. *"Imagenet classification with deep convolutional neural networks."* Advances in neural information processing systems. 2012.

[4] Wan, Li, et al. *"Regularization of neural networks using dropconnect."* Proceedings of the 30th International Conference on Machine Learning (ICML-13). 2013.

[5] Hinton, Geoffrey E., et al. *"Improving neural networks by preventing co-adaptation of feature detectors."* arXiv preprint arXiv:1207.0580 (2012).

[6] Chih-Chung Chang and Chih-Jen Lin, *"LIBSVM – A Library for Support Vector Machines"* `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`