CS7015: DEEP LEARNING
Assignment 3 Report

Authors:
Pranav Gadikar: CS16B115
Soham Dixit: CS16B006
Rahul Chakwate: AE16B005

May 8, 2019

Contents

1 Task1: Image classification using a MLFFNN with Deep CNN features for an image as the input to the MLFFNN

1.1 Final Model

The size of the input feature vector to the MLFFNN is 512*7*7=25088. The network consists of 2 hidden layers, each having 4096 nodes each followed by the output layer.

Using VGGNet as a feature extractor:

Table 1: Training Accuracy: 64.4%, Test Accuracy:64.5%

Training data Confusion					Test data Confusion				
binoculars	motorbikes	people	watch	clutter	binoculars	motorbikes	people	watch	clutter
70.86%	7.9%	1.3%	5.2%	19.2%	64.6%	9.2%	3%	7.6%	15.3%
14.8%	63.2%	3.5%	4.1%	14.1%	15%	62.9%	2.1%	2.1%	17.9%
4.8%	8.2%	70.5%	4.8%	11.7%	3.1%	7.9%	74.6%	1.6%	15.8%
9.2%	3.5%	2.8%	67.1%	17.1%	9.8%	6.5%	3.2%	67.2%	13.1%
11.3%	11.5%	3.9%	17.8%	55.2%	11.2%	12.9%	4.8%	13.7%	57.2%

Using GoogLeNet as a feature extractor:

Table 2: Training Accuracy:68.8%, Test Accuracy:68.5%

Training data Confusion					Test data Confusion				
binoculars	motorbikes	people	watch	clutter	binoculars	motorbikes	people	watch	clutter
74.8%	6.6%	2.6%	4.6%	11.2%	69.2%	6.1%	3%	10.7%	10.7%
10.3%	69.3%	3%	3.5%	13.6%	9.6%	70.8%	2.5%	2.1%	15%
4.1%	6.8%	78.7%	3.4%	6.8%	4.7%	9.4%	69.8%	4.7%	11.1%
7.1%	5.7%	2.8%	70%	14.2%	9.8%	4.9%	6.5%	72.1%	6.5%
10%	10.3%	2.9%	12.4%	64.2%	9.2%	12.5%	2%	8.4%	67.7%

1.2 Observations:

Though both the accuracies are not that high, GoogleNet based features are better at clutter identification and in general perform better than VGGNet based features on the dataset.

2 Task2: Image annotation using Deep CNN features as input to MLFFNN:

2.1 Data sizes:

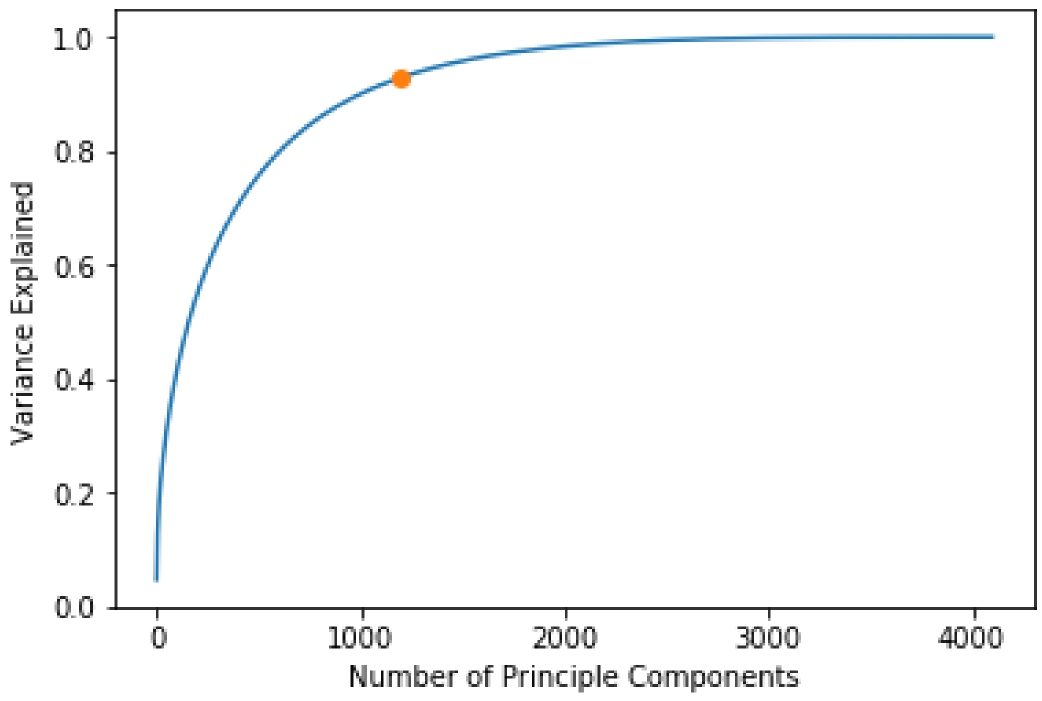
Train Data: 7000 Images
Test Data: 3000 Images
No. of labels: 6

2.2 Dimension reduction using PCA:

Pre-final layer of VGG-net is 4096 dimensional and that of GoogLe-net is 1024 dimensional leading to large overfitting of the data.

PCA is applied on both GoogLe-net and VGG-net to get their reduced dimension representation.

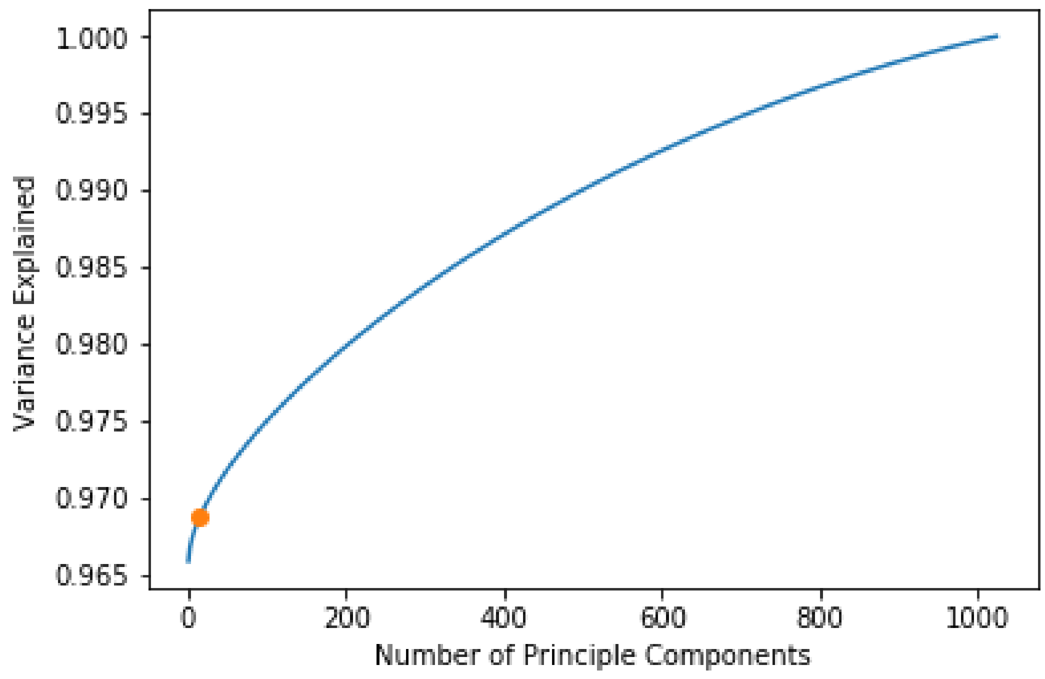
2.3 PCA for VGG-net:



(1).png

Reduced dimension is chosen to be 1200.

2.4 PCA for GoogLe-net:



(2).png

Reduced dimension is chosen to be 15.

2.5 Experiments with VGG-net Features:

Table 3: Experiments for different number of nodes in hidden layers of MLFFN:

Hidden-1 Nodes	Hidden-2 Nodes	Epochs	Loss
120	80	30	0.0217
120	50	37	0.0218
150	80	30	0.0212
80	50	45	0.0206

Best Model for VGG-net = 80,50

2.6 Experiments with GoogLe-net Features:

Table 4: Experiments for different number of nodes in hidden layers of MLFFN:

Hidden-1 Nodes	Hidden-2 Nodes	Epochs	Loss
20	10	23	0.1810
20	20	18	0.1809
10	15	26	0.1810

Hidden-1 Nodes	Hidden-2 Nodes	Epochs	Loss
120	80	30	0.0217

Best Model for GoogLe-net = 10,15

2.7 Comparing Accuracies for various labels for VGG-net and GoogLe-net:

Table 5:

GoogLe-net Accuracy						VGG-net Accuracy					
Animal	Window	Plant	Cloud	Water	Bldg	Animal	Window	Plant	Cloud	Water	Bldg
79.5%	84.2%	85.8%	65.8%	51.4%	74.3%	69.3%	77.5%	78.6%	55.3%	49.2%	63.8%

Best Model accuracy for VGG-net = 67%
Best Model accuracy for GoogLe-net = 74%

2.8 Observation and Inferences:

- Google net performs better than VGG net by extracting better global features
- Google net takes much lesser time to extract features than VGG net.
- Google net features requires lesser parameters to be estimated for MLFFN.

3 Task4: Image captioning using VLAD features and LSTM:

3.1 Introduction:

Teacher Forcer Method was used while training the model which is used by the DL practitioners in most scenarios.

The teacher forcer algorithm steps:

1. Feed a <start> (start of a sentence or a word) token to the LSMT cell as an input at time t=0.
2. Find the index of the most probable character (word) at time t=0, using argmax.
3. Feed the next token (next embedded word from our target) to the LSMT cell as an input at time t=1.
4. Repeat until <end> token is obtained as the output of the cell.

We do not feed the last most probable word but we feed the already available next word embedding.

3.2 Data sizes and Dimensions:

To find the best model, experiments were conducted on a subset of the dataset containing 1400 images for training and next 600 images for validation.
The configuration of model was as follows:

- Output of VGG16 = N*512*7*7 (N*dimensions*kernelsize)
- Input to VLAD model = N*512*7*7
- Output of VLAD = N*(K*d)
- Input to decoder LSTM = features+target captions
- Output of decoder LSTM = one hot vector of prediction captions
- Loss = CrossEntropyLoss
- Optimizer = Adam(lr=0.001)
- Embedding = torch.nn.Embedding
- BLEU score = BLEU 4 score

3.3 Experiments:

3.4 Changing number of clusters:

Variation of the Bleu score on the toy dataset was explored with change in number of clusters with values 3,8 and 16. However no great improvement was observed in return of the increased computational complexity.

3.5 Changing number of Hidden layers:

Table 6: Experiments for different number of nodes in hidden layers of VGG-net VLAD:

Hidden Nodes	BLEU-4 Train	BLEU-4 Test
40	13.73	11.6
20	10.93	9.1
10	9.1	7.8

Table 7: Experiments for different number of nodes in hidden layers of Google-net VLAD:

Hidden Nodes	BLEU-4 Train	BLEU-4 Test
40	12.65	11.85
20	10.03	8.14
10	8.9	7.6

Best Model BLEU-4 score for VGG-net = 11.6% on Test
Best Model BLEU-4 score for GoogLe-net = 11.85% on Test

3.6 Observation and Inferences:

- As K clusters increases, the BLEU4 train and validation also increases. The increase is very slow beyond a certain point.
- Google-net and VGG-net have almost same performance.
- BLEU score increases with increase in no. of hidden layers.

4 Task5: Video Captioning using ActionVLAD as encoder and a single hidden layer LSTM based RNN as decoder.

Youtube Clips and corresponding captions were used as the dataset. Experiments performed include:

- Changing Encoder CNN Architecture
- Changing K no. of Clusters
- Changing no. of nodes in hidden layer of decoder LSTM network

Teacher Forcer Method was used while training the model which is used by the DL practitioners in most scenarios.

The teacher forcer algorithm steps:

1. Feed a <start> (start of a sentence or a word) token to the LSMT cell as an input at time t=0.
2. Find the index of the most probable character (word) at time t=0, using argmax.
3. Feed the next token (next embedded word from our target) to the LSMT cell as an input at time t=1.
4. Repeat until <end> token is obtained as the output of the cell.

We do not feed the last most probable word but we feed the already available next word embedding.

4.1 Experiments with changing K in clusters

To find the best model, experiments were conducted on a toy dataset containing 100 videos for training and next 50 videos for validation.
The configuration of model was as follows:

- Output of VGG16 = N*30*512*7*7 (N*framecount*dimensions*kernelsize)
- Input to ActionVLAD model = N*30*512*7*7
- Output of ActionVLAD = N*(K*d)
- Input to decoder LSTM = features+target captions
- Output of decoder LSTM = one hot vector of prediction captions
- Loss = CrossEntropyLoss
- Optimizer = Adam(lr=0.001)
- Embedding = torch.nn.Embedding
- BLEU score = BLEU 4 score

The BLEU 4 scores are reported below.

Table 8: Experiments for different no. of clusters:

K	BLEU4 Train	BLEU4 Val
4	35.62	9.65
8	37.27	9.93
16	37.58	9.96

4.2 Experiments with no. of Nodes in the hidden layer of the decoder LSTM network

Keeping K=8, no. of Nodes in hidden Layer are varied as follows.

Table 9: Experiments for different no. of nodes in Hidden Layer of LSTM Cell:

No. of Nodes	BLEU4 Train	BLEU4 Val
32	32.46	8.34
64	37.27	9.73

4.3 Final Models with different Encoder Architectures

For the final model, K=8 and no. of hidden nodes = 64 is chosen.
Entire training data of 600 videos is used for training. 100 videos for validation and 300 videos for testing.
BLEU 4 scores for models with VGG and GoogleNet encoder are given below.

Table 10: BLEU 4 Scores for different Encoder Architecture

Model	Train	Validation	Test
VGG16	14.92	13.76	12.63
GoogleNet	14.48	13.71	12.5

NOTE: The State of Art BLEU 4 score as of 2018 is about 35.
Show and tell[3] BLEU4 score=27.7
Show attend and tell[4] BLEU4 score = 20-22, BLEU1 score = 66-71

4.4 Observations and Conclusion

- As K clusters increases, the BLEU4 train and validation also increases. The increase is very slow beyond a certain point.
- When no. of Nodes is increased from 32 to 64, BLEU4 Score increases as the parameters increases and the model fits well to the data.
- VGG16 and GoogleNet performs at par with each other.

- The final BLEU4 score is about 13 for the test dataset. Given that the train dataset was small, this value is comparable to some of the novel papers like 'Show and tell[3]' and 'Show, attend and tell[4]'.

REFERENCES

[1] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, Josef Sivic NetVLAD: CNN architecture for weakly supervised place recognition. CVPR Paper

[2] Rohit Girdhar, Deva Ramanan, Abhinav Gupta ActionVLAD: Learning spatio-temporal aggregation for action classification. In CVPR 2017.

[3] Oriol Vinyals, Alexander Toshev, Samy Bengio. Show and Tell: A Neural Image Caption Generator. In CVPR 2015.

[4] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In ICML 2015.