CS698O: Mid Term Results Report
Modified Stacked Attention Networks for VQA using Visual grounding of Phrases

Group-23 : Abhishek Verma(14026) , Ayush Tulsyan(14167)

November 25, 2017

1 Results

We tested our model on MS-COCO validation dataset for visual question answering. Without using diminishing learning rate, accuracy was around 32% while with diminishing learning rate in place, accuracy increased to 44%.

2 Inferences

- Training error decreases and accuracy increases with an increase in number of lSTM layers till a point after which it starts increasing. [Figure1, Figure2]

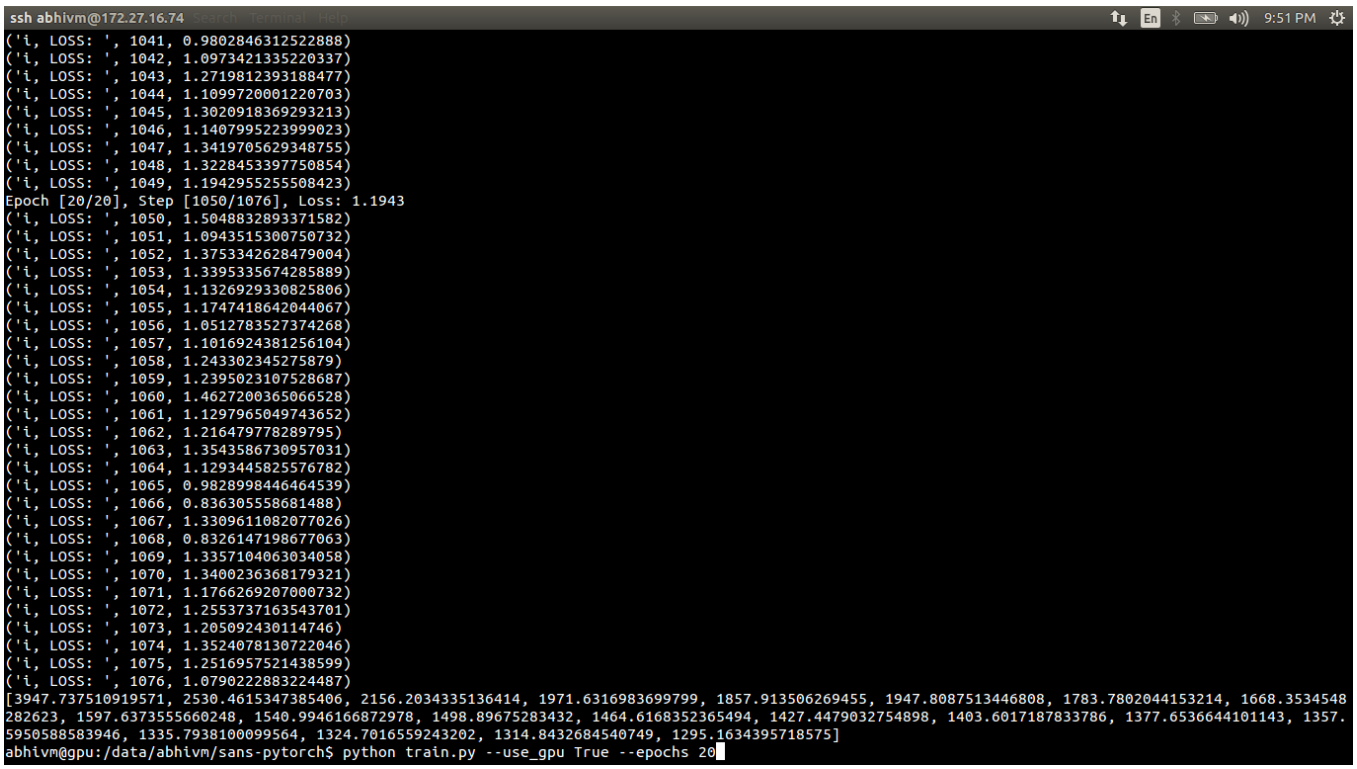

Figure 1: Train errors with LSTM of single layer

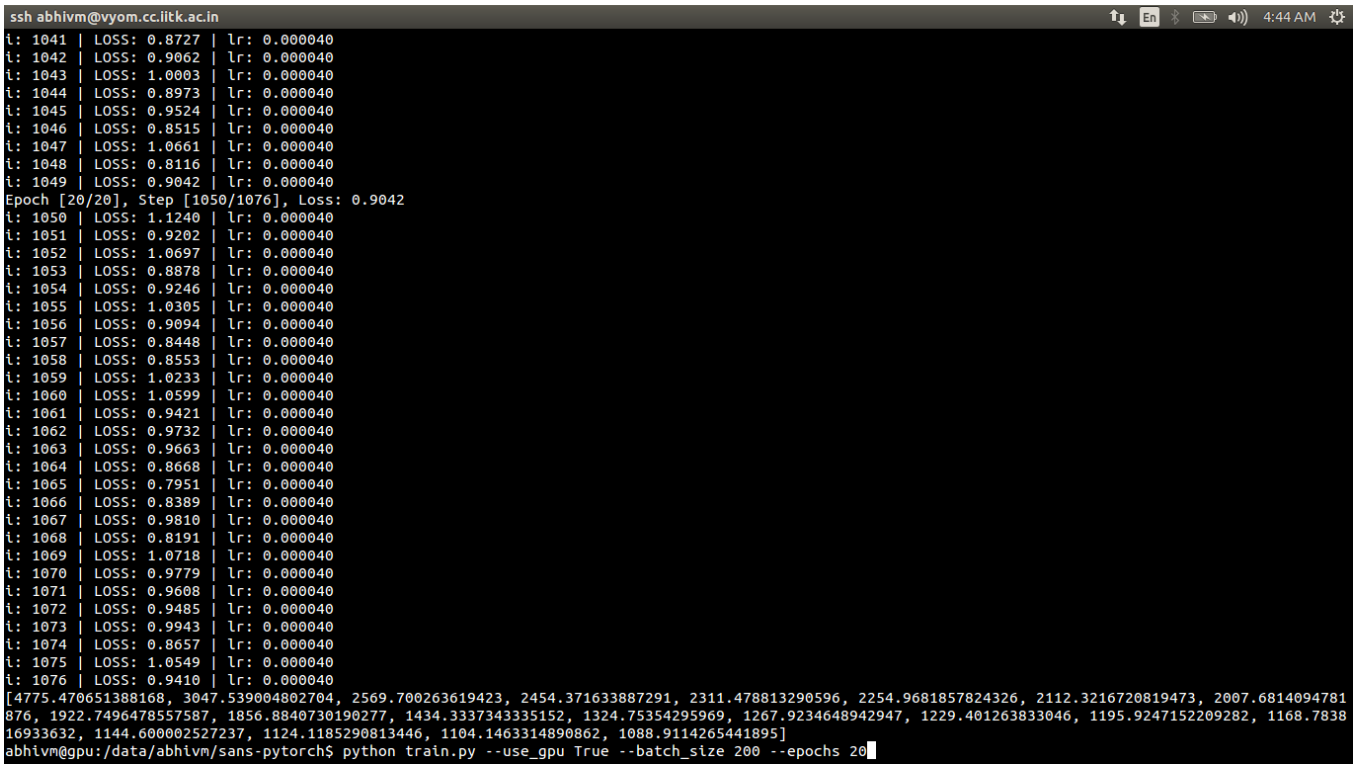

Figure 2: Train errors with LSTM of double layer

- Training error decreases when number of attention layers were increased from one to two. However it started increasing as the number of attention layers were increased further from two. The case with accuracy is vice versa. [Figure3]
- Training error started increasing after 25th epoch when we kept the learning rate constant. We analyzed it and started diminishing the learning rate at as the epochs proceeded. This resulted in an almost constant training error after 30th iteration. [Figure4]
- Variation of learning rate with number of epochs. [Figure5, Figure6, Figure7]
- Variation of accuracy with learning rate(with diminishing learning rate in place). [Figure8, Figure9, Figure10, Figure11]
- Variation of accuracy with number of epochs. [Figure12, Figure13, Figure14, Figure15, Figure16, Figure17, Figure18, Figure19, Figure20, figure21, Figure22]

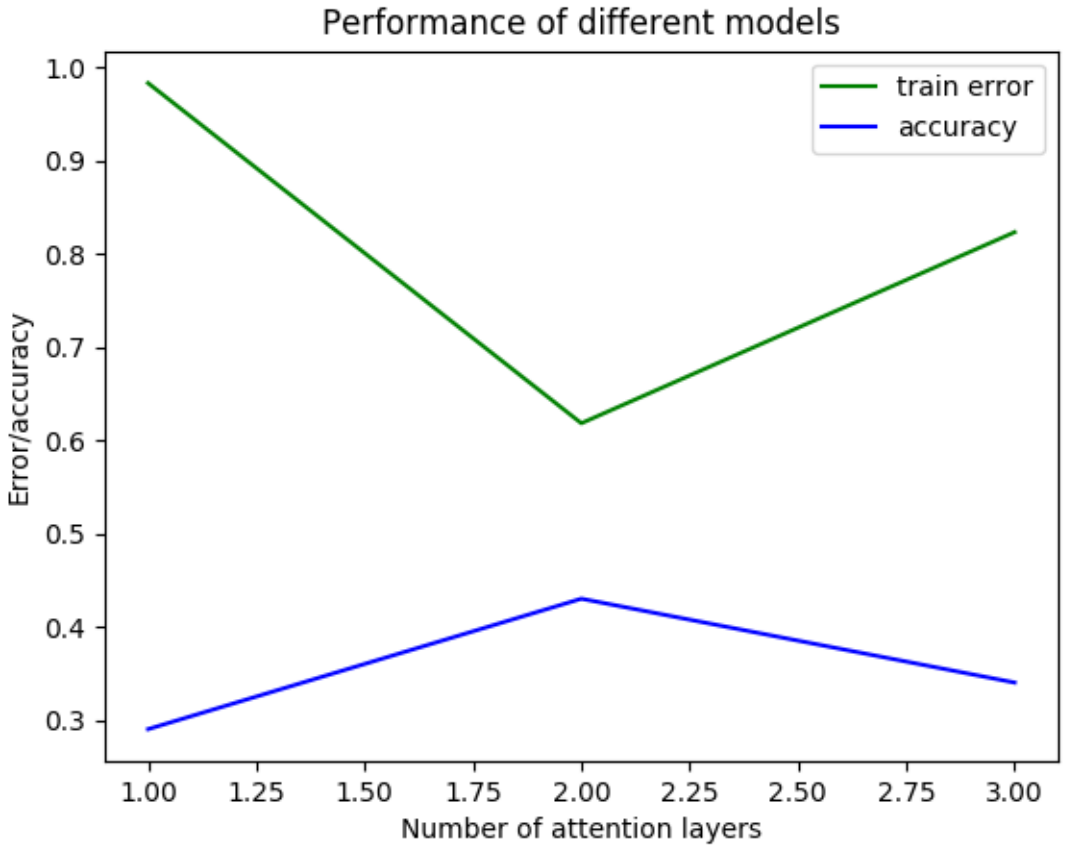

Figure 3: Performance of different models

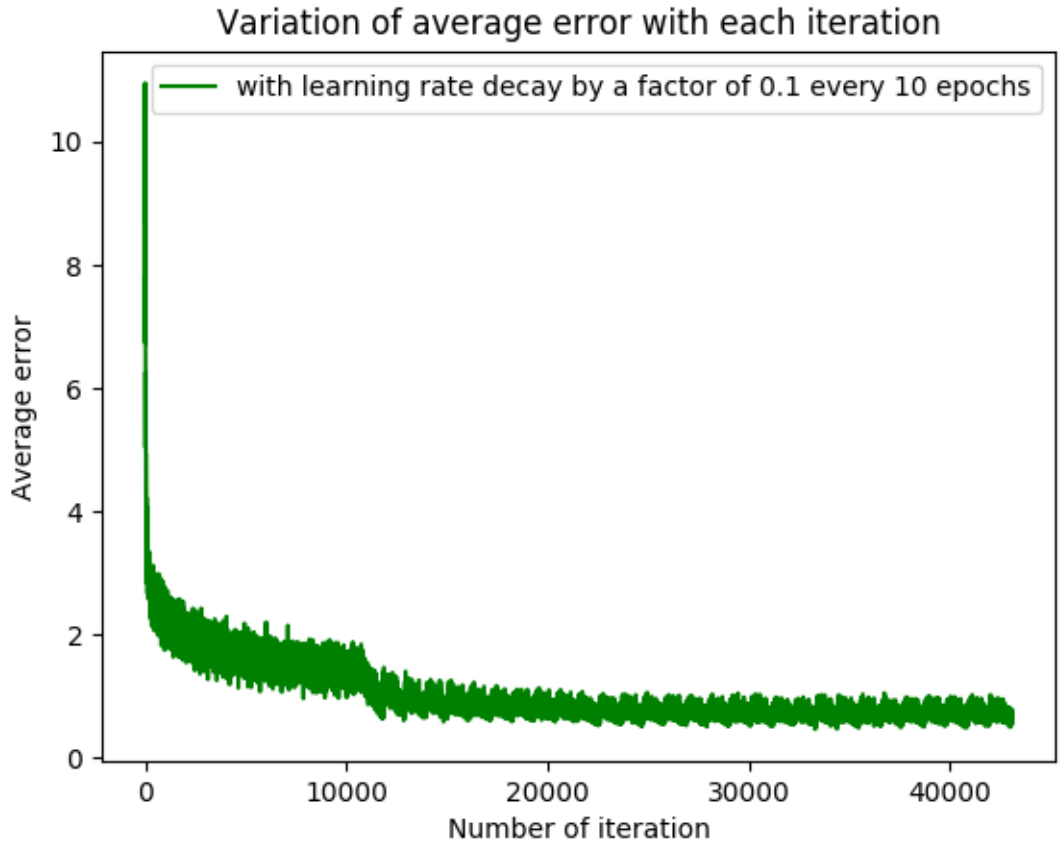

Figure 4: Variation of training error

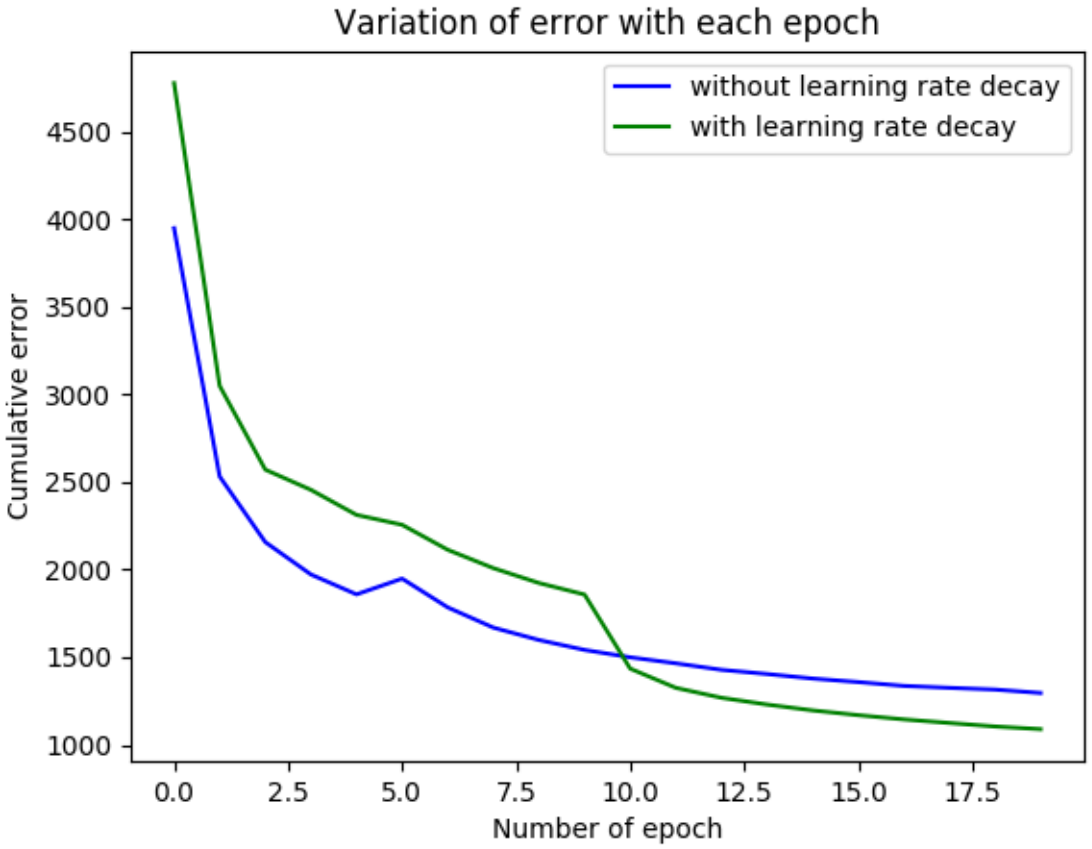

Figure 5: Variation of training error with number of epochs

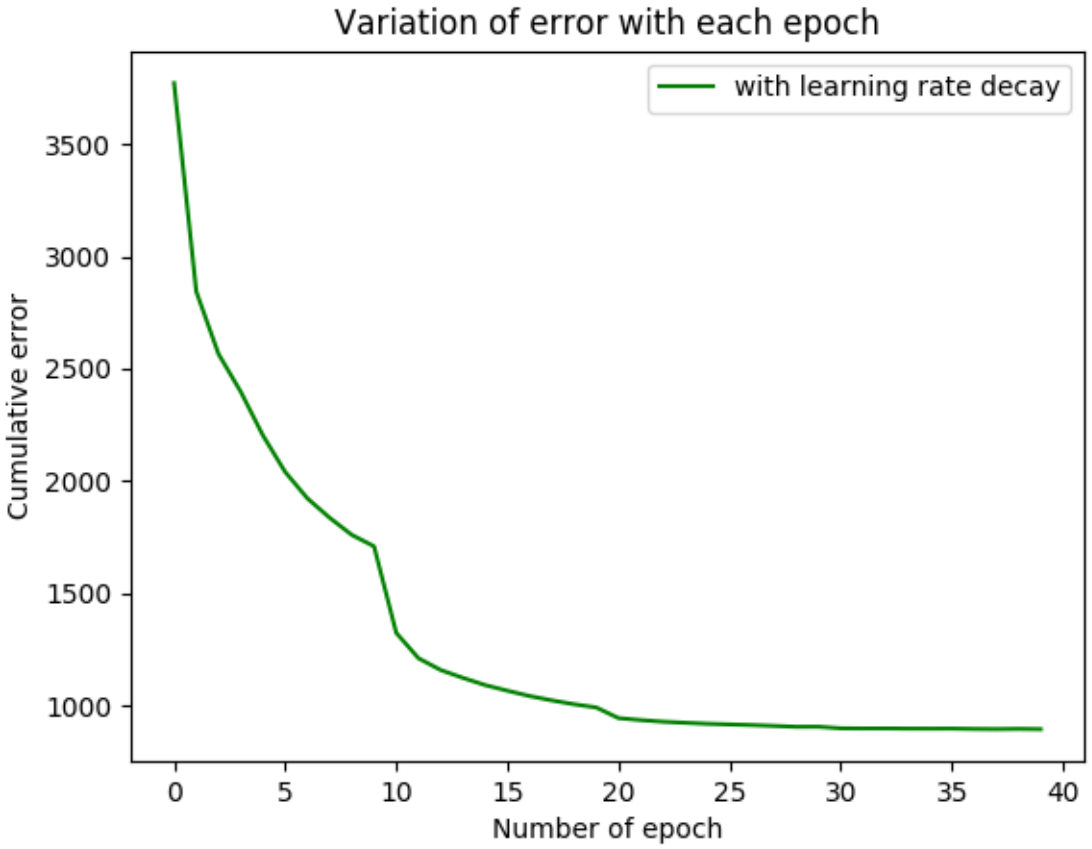

Figure 6: Variation of training error with number of epochs

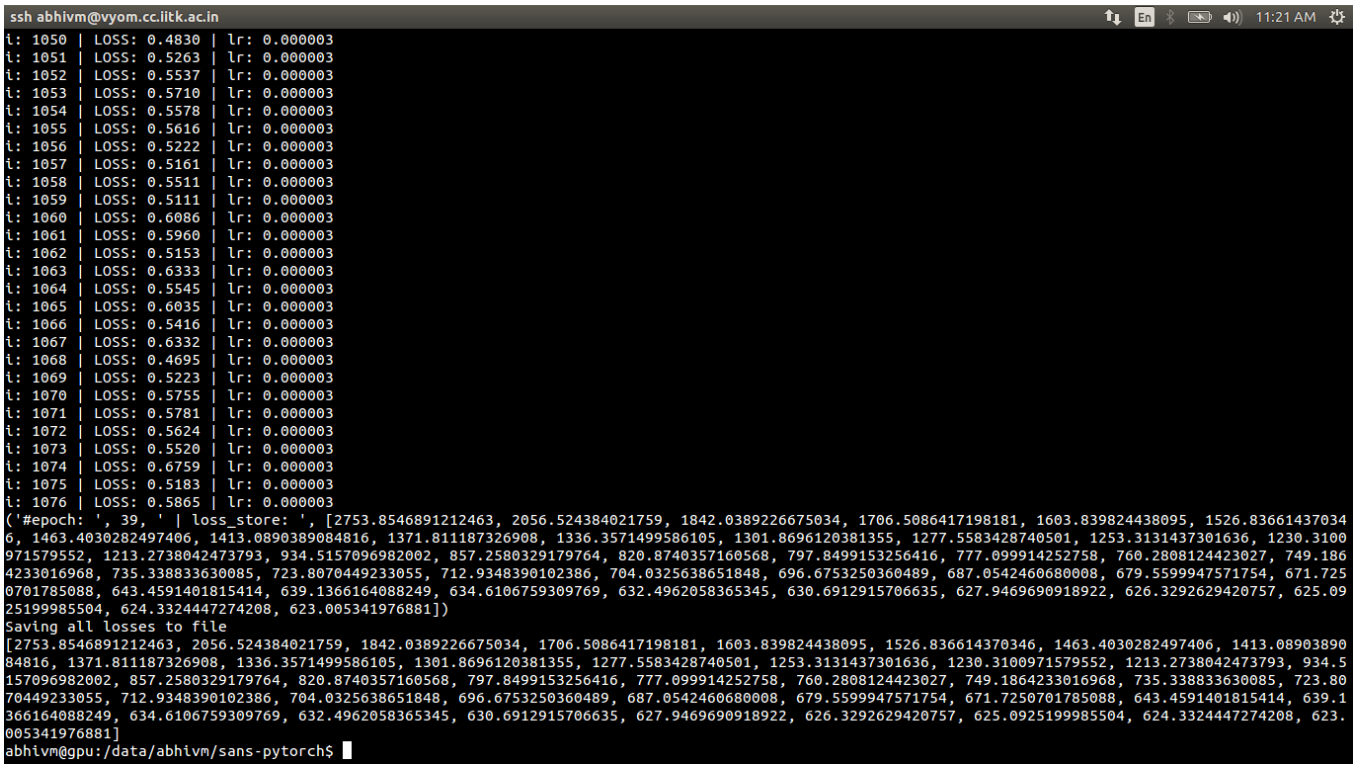

Figure 7: Variation of training error with number of epochs

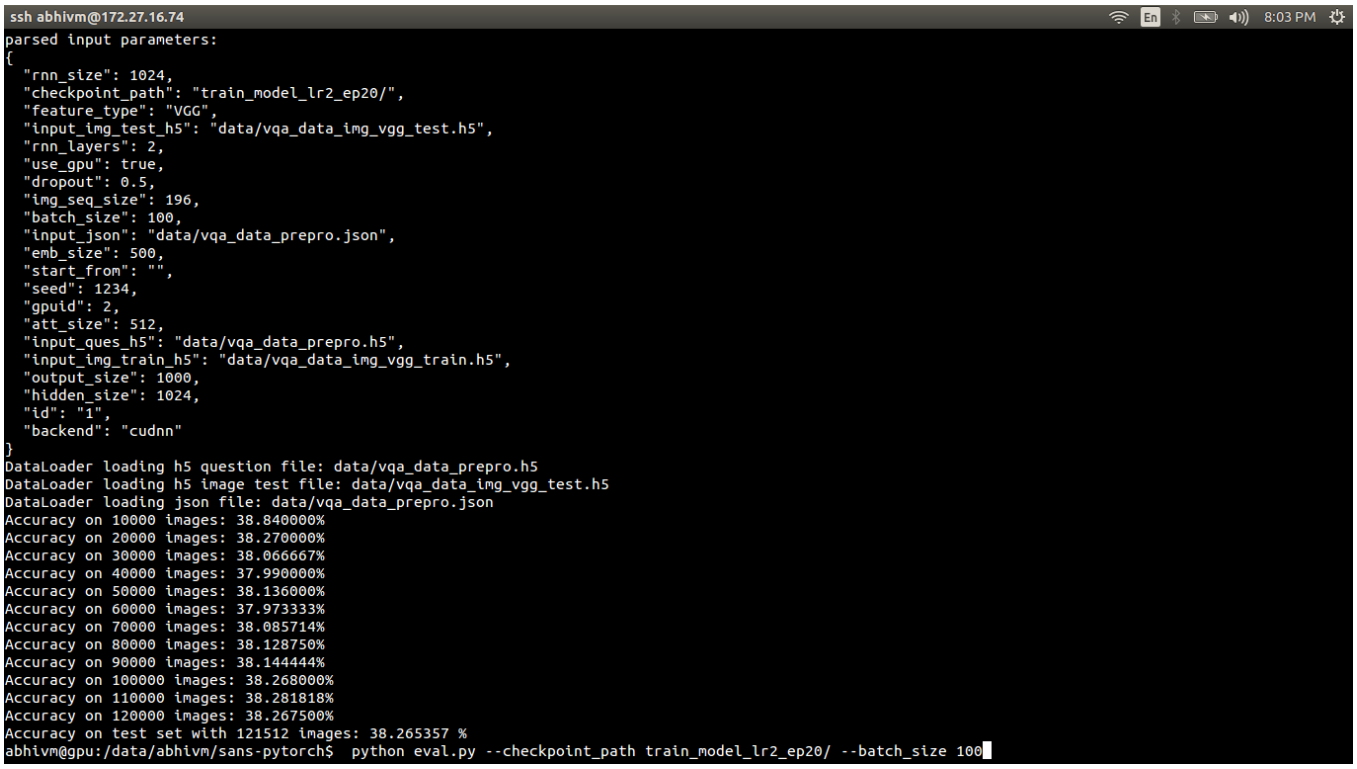

Figure 8: Accuracy with learning rate=0.0002

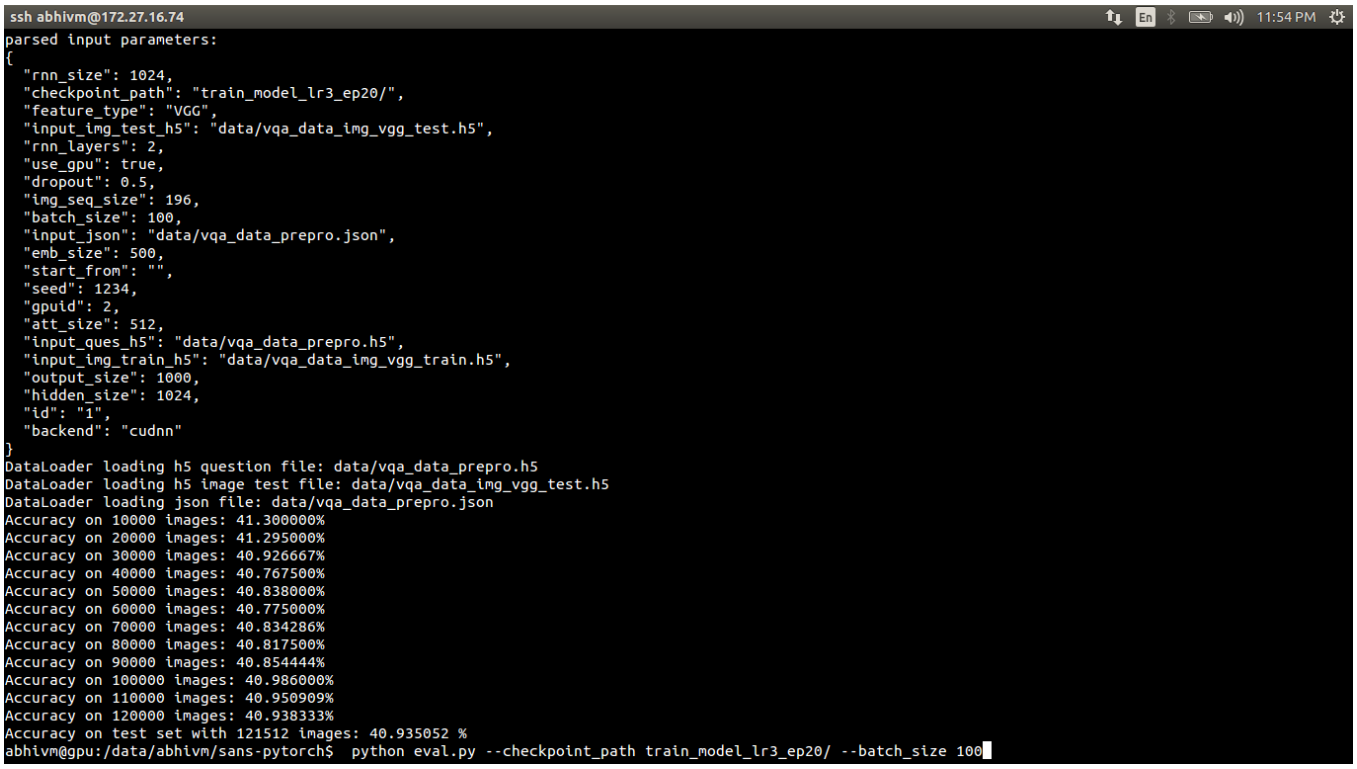

Figure 9: Accuracy with learning rate=0.0003

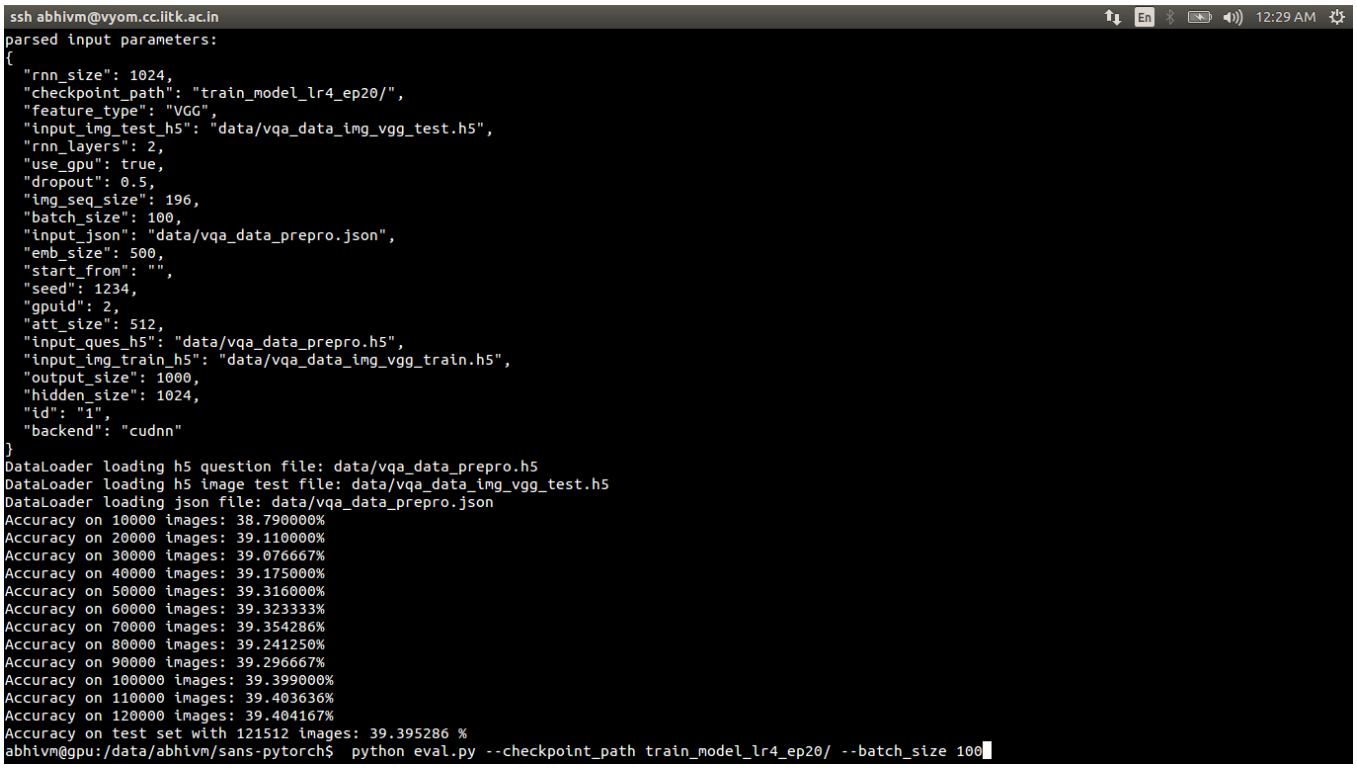

Figure 10: Accuracy with learning rate=0.0004

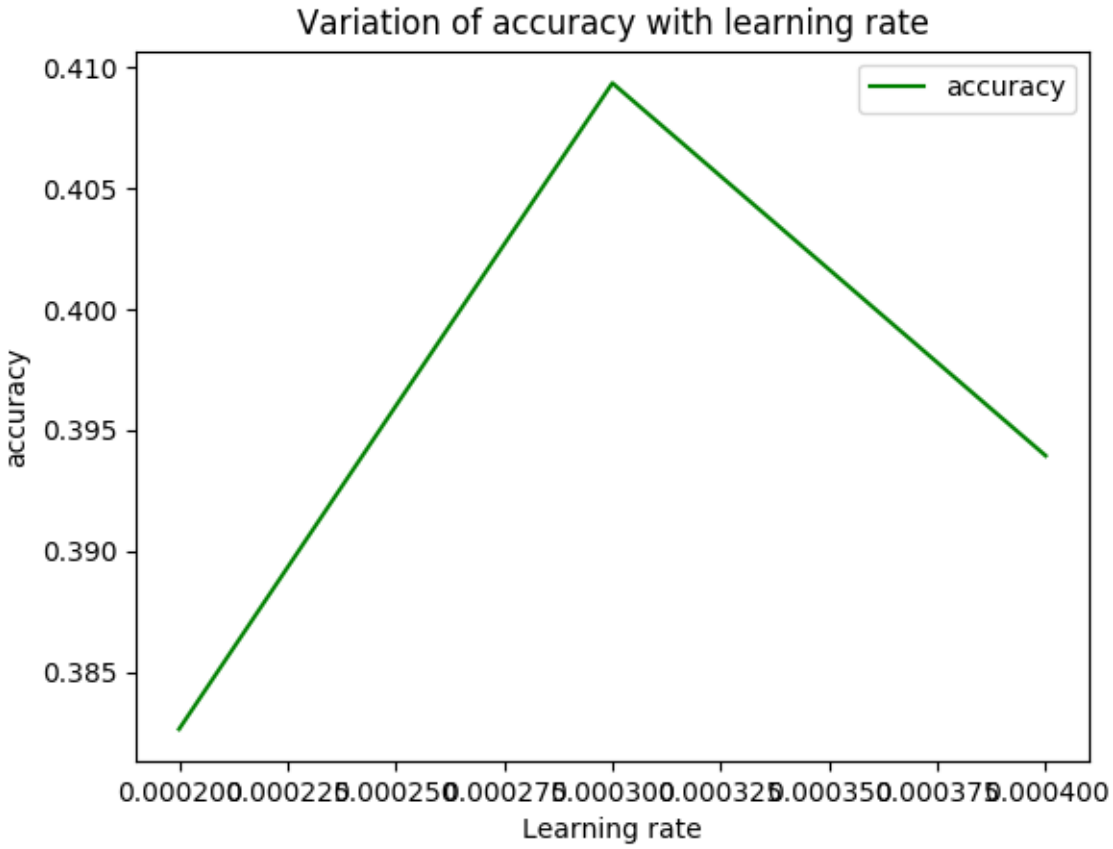

Figure 11: Variation of accuracy with learning rate

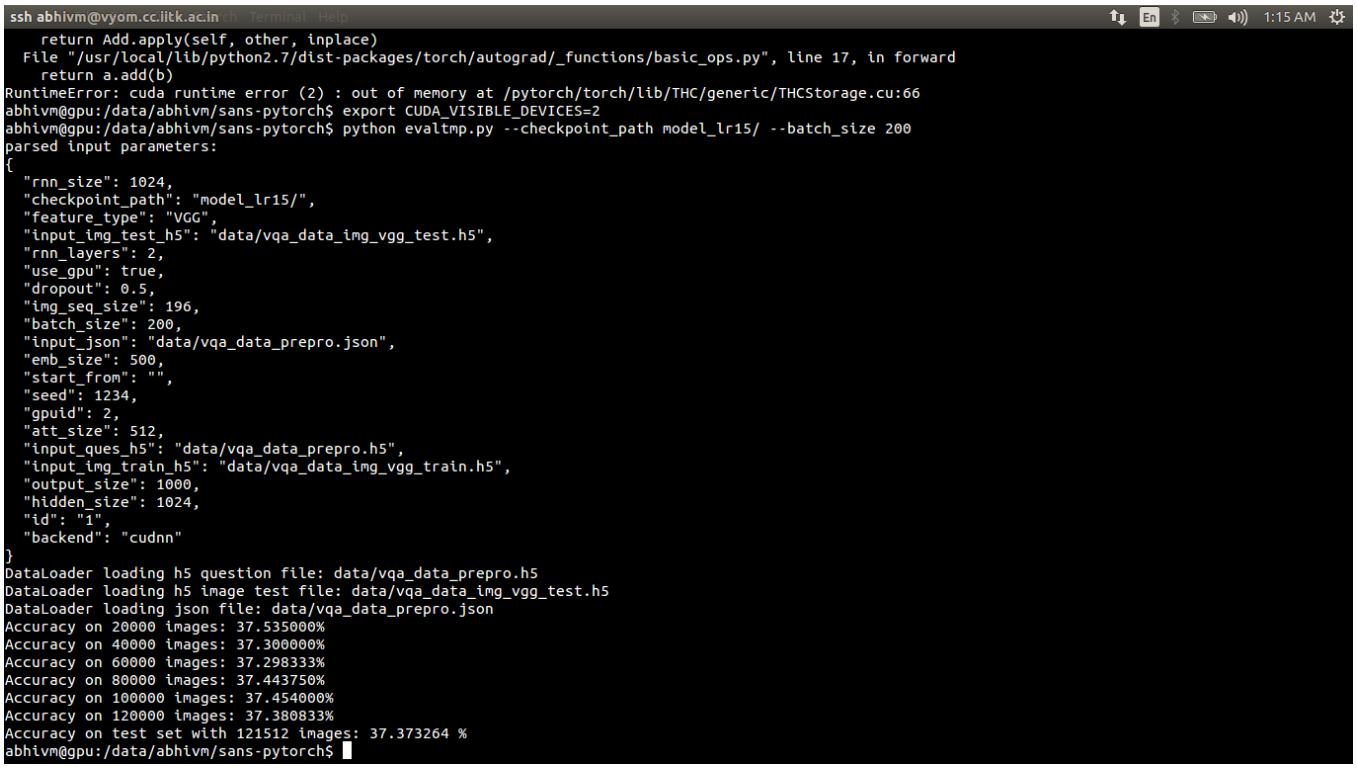

Figure 12: Accuracy after 1st epoch

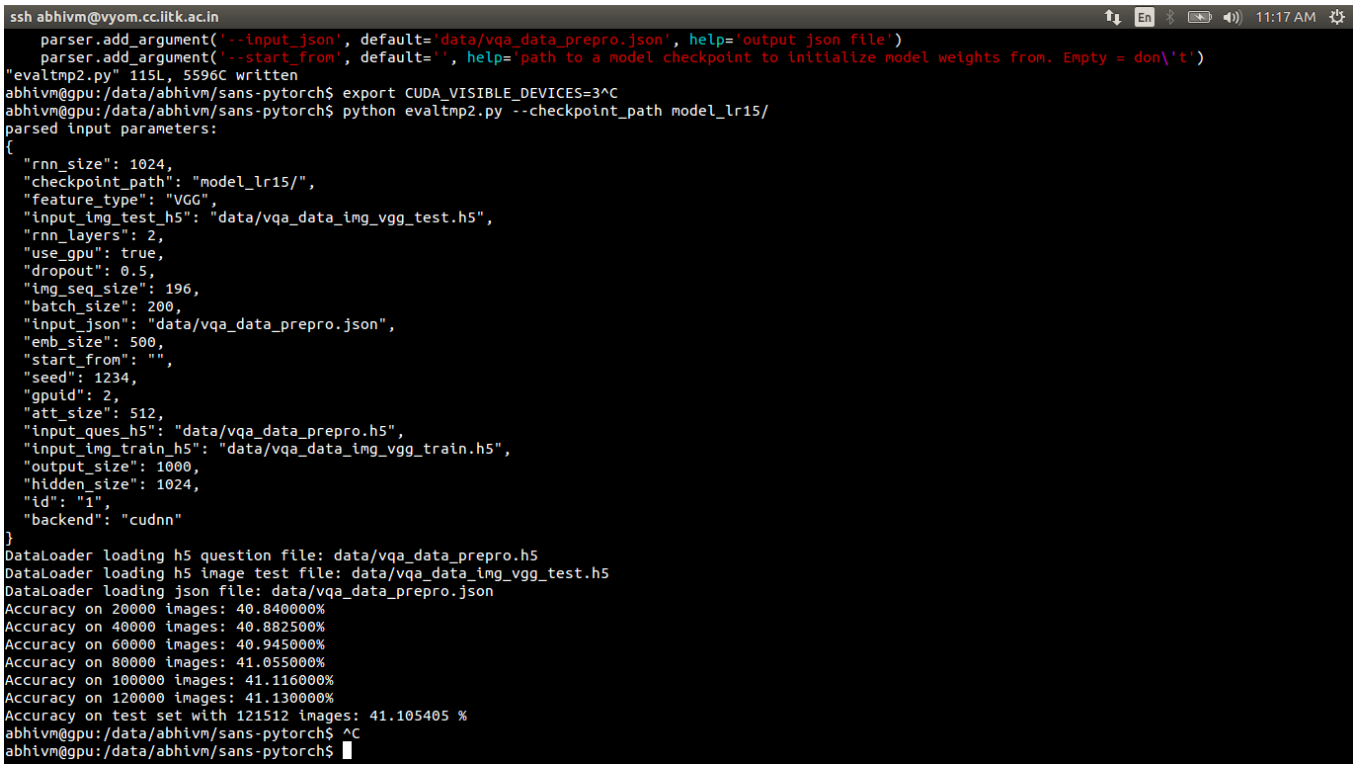

Figure 13: Accuracy after 9th epoch

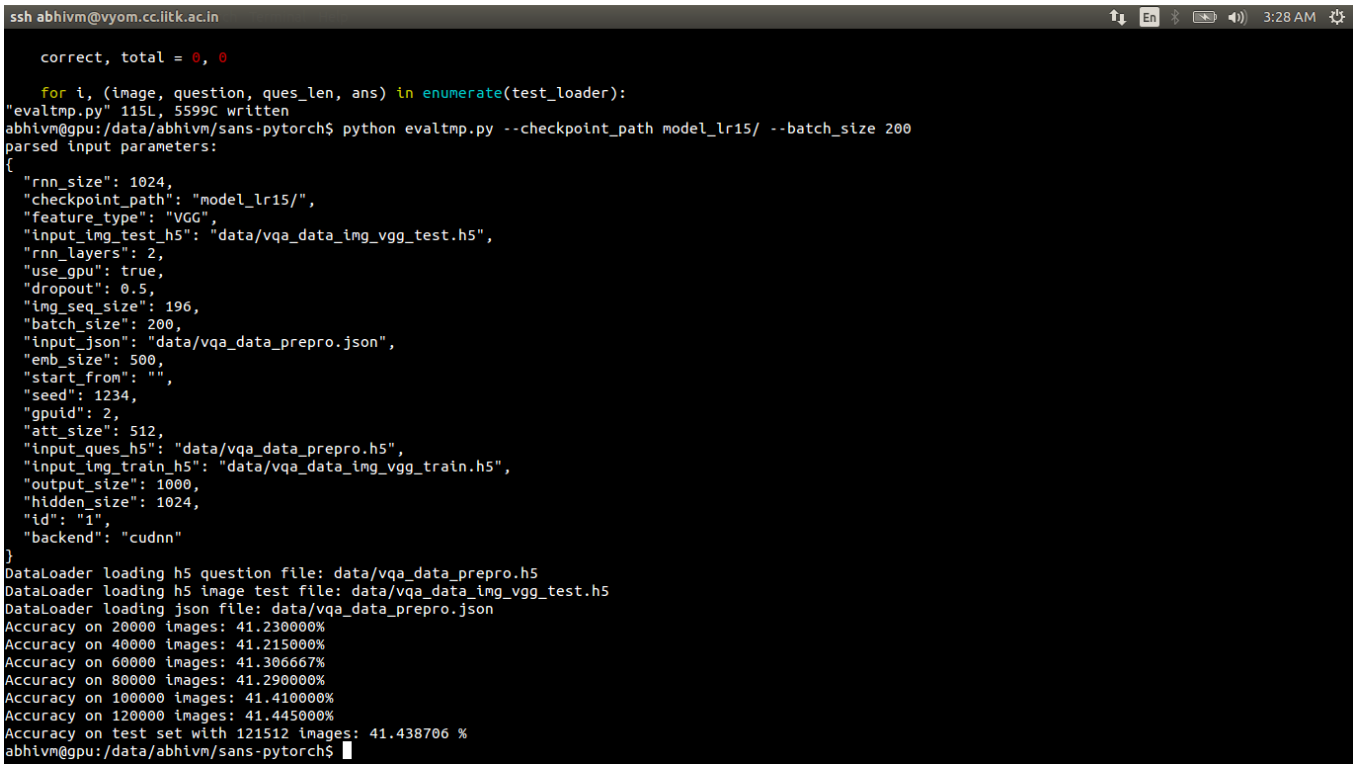

Figure 14: Accuracy after 11th epoch

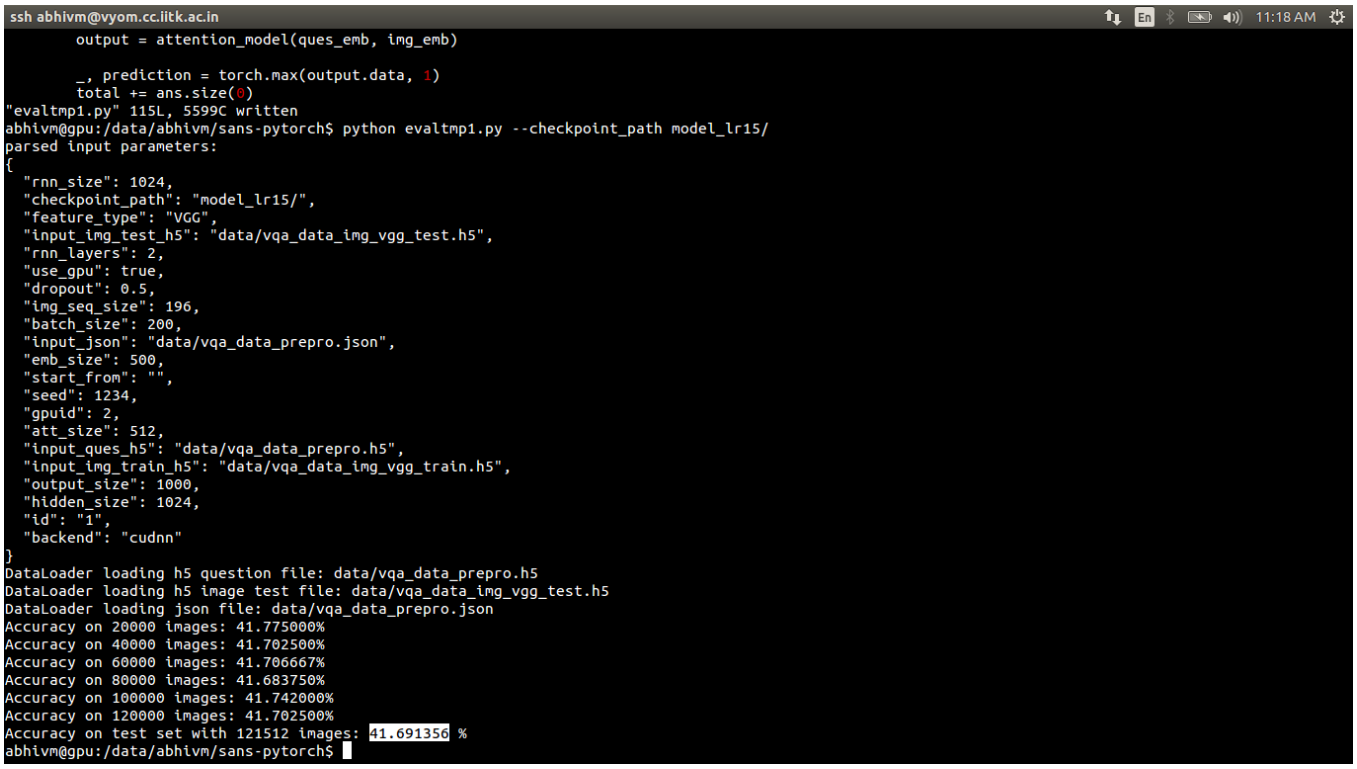

Figure 15: Accuracy after 14th epoch

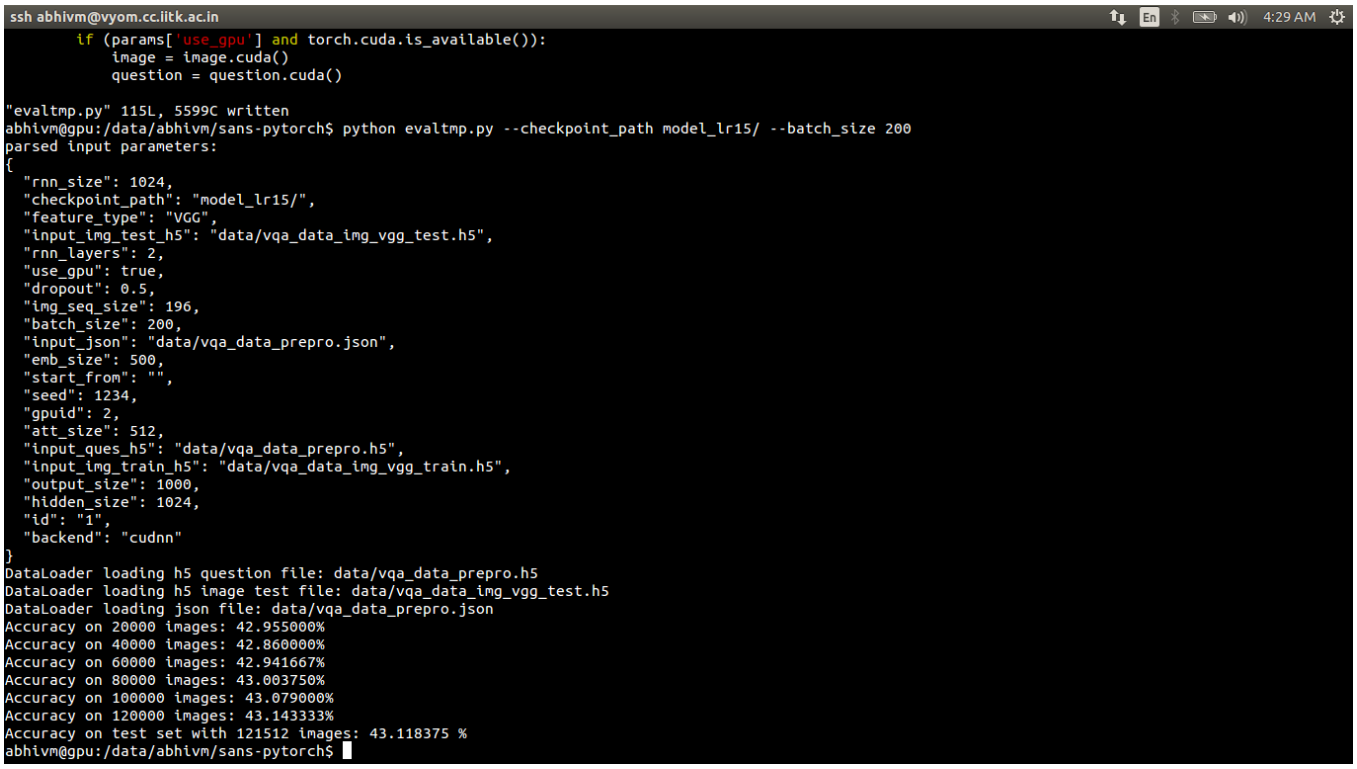

Figure 16: Accuracy after 16th epoch

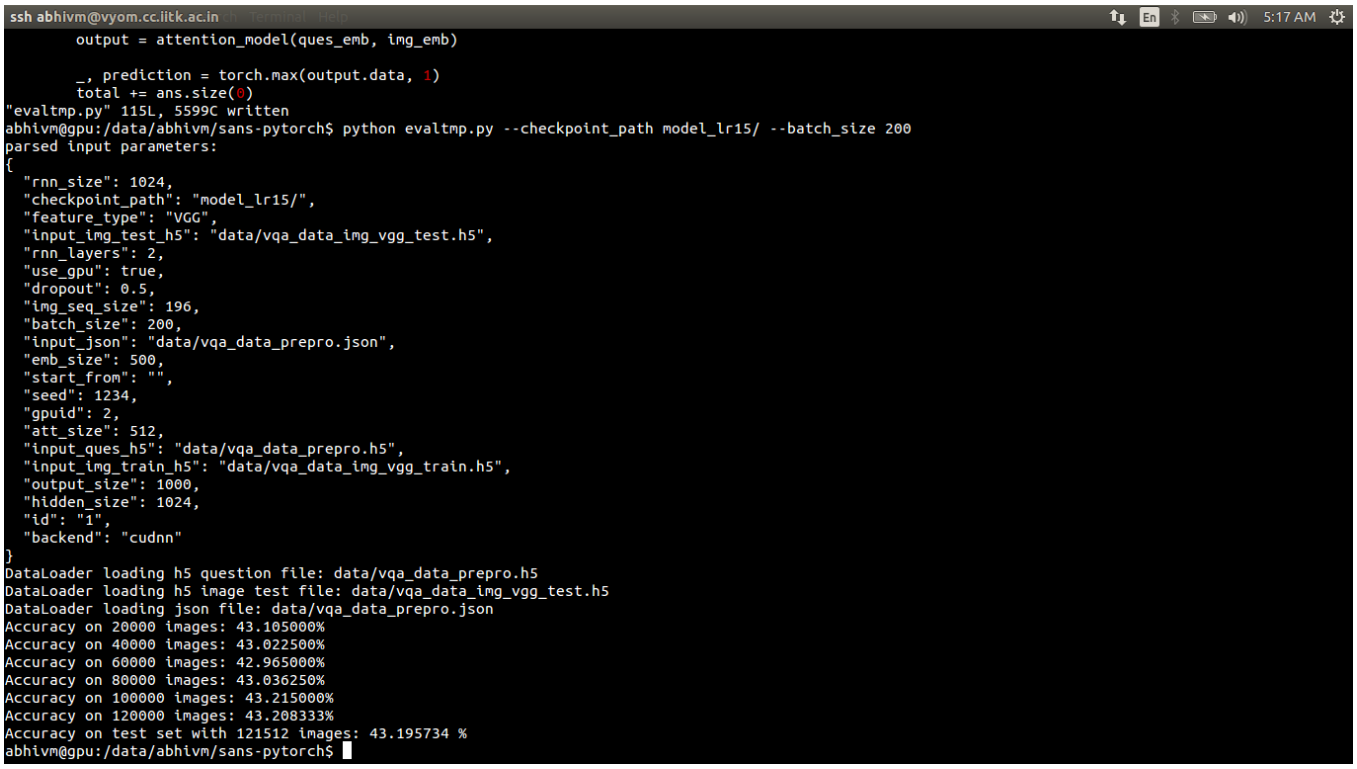

Figure 17: Accuracy after 21st epoch

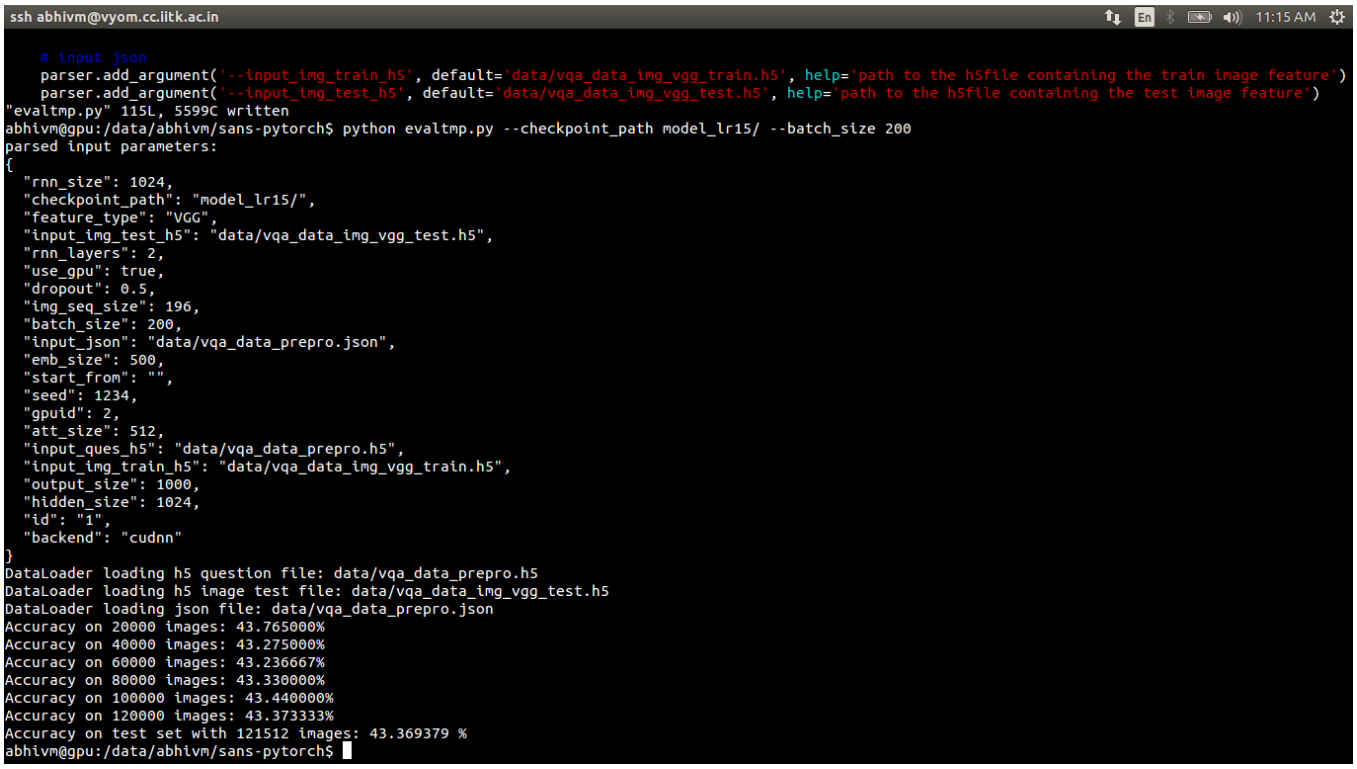

Figure 18: Accuracy after 24th epoch

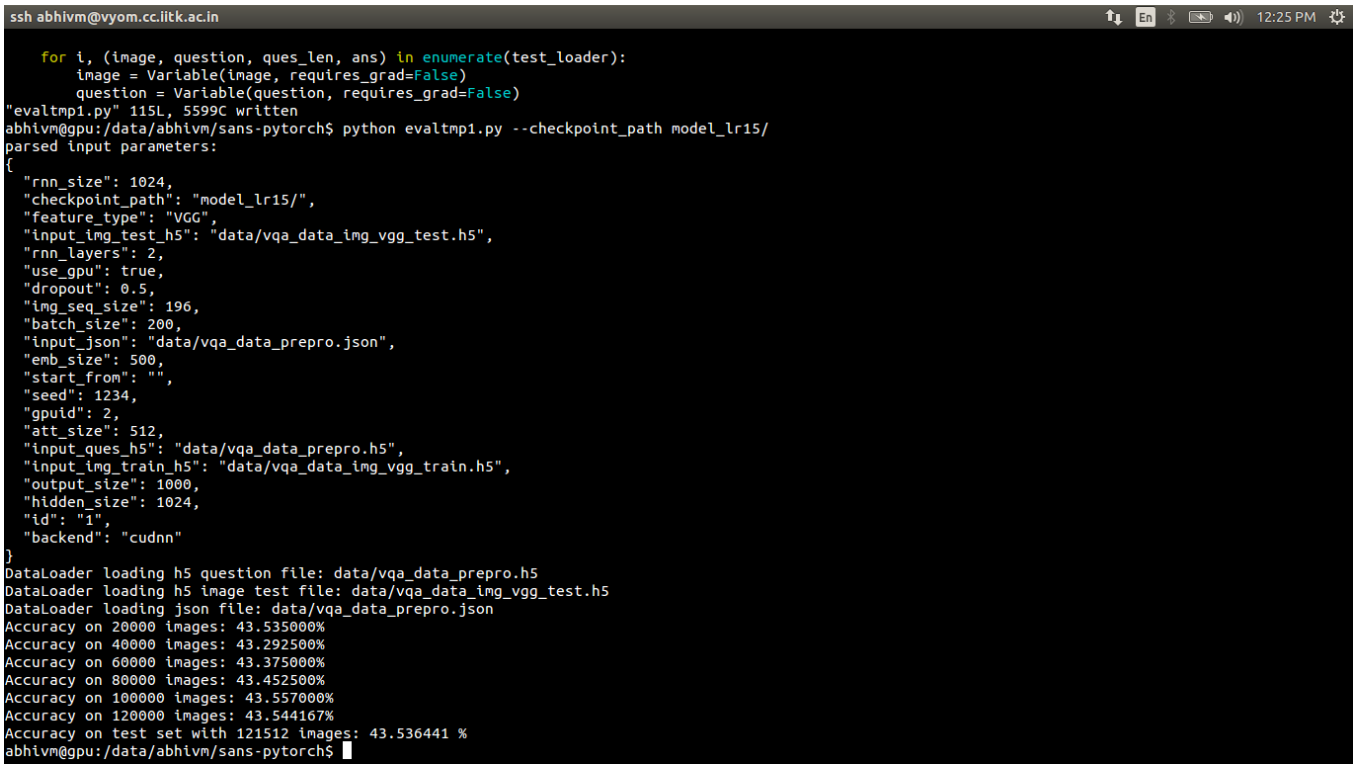

Figure 19: Accuracy after 31st epoch

```
ssh abhivm@vyom.cc.iitk.ac.in                                                    12:48 PM
            image = image.cuda()
            question = question.cuda()

        img_emb = image_model(image)
"evaltmp2.py" 115L, 5599C written
abhivm@gpu:/data/abhivm/sans-pytorch$ python evaltmp2.py --checkpoint_path model_lr15/
parsed input parameters:
{
  "rnn_size": 1024,
  "checkpoint_path": "model_lr15/",
  "feature_type": "VGG",
  "input_img_test_h5": "data/vqa_data_img_vgg_test.h5",
  "rnn_layers": 2,
  "use_gpu": true,
  "dropout": 0.5,
  "img_seq_size": 196,
  "batch_size": 200,
  "input_json": "data/vqa_data_prepro.json",
  "emb_size": 500,
  "start_from": "",
  "seed": 1234,
  "gpuid": 2,
  "att_size": 512,
  "input_ques_h5": "data/vqa_data_prepro.h5",
  "input_img_train_h5": "data/vqa_data_img_vgg_train.h5",
  "output_size": 1000,
  "hidden_size": 1024,
  "id": "1",
  "backend": "cudnn"
}
DataLoader loading h5 question file: data/vqa_data_prepro.h5
DataLoader loading h5 image test file: data/vqa_data_img_vgg_test.h5
DataLoader loading json file: data/vqa_data_prepro.json
Accuracy on 20000 images: 43.865000%
Accuracy on 40000 images: 43.380000%
Accuracy on 60000 images: 43.433333%
Accuracy on 80000 images: 43.466250%
Accuracy on 100000 images: 43.549000%
Accuracy on 120000 images: 43.511667%
Accuracy on test set with 121512 images: 43.489532 %
abhivm@gpu:/data/abhivm/sans-pytorch$
```

Figure 20: Accuracy after 36th epoch

```
ssh abhivm@vyom.cc.iitk.ac.in                                                    12:26 PM
        image = Variable(image, requires_grad=False)
        question = Variable(question, requires_grad=False)
        if (params['use_gpu'] and torch.cuda.is_available()):
            image = image.cuda()
"evaltmp2.py" 115L, 5599C written
abhivm@gpu:/data/abhivm/sans-pytorch$ python evaltmp2.py --checkpoint_path model_lr15/
parsed input parameters:
{
  "rnn_size": 1024,
  "checkpoint_path": "model_lr15/",
  "feature_type": "VGG",
  "input_img_test_h5": "data/vqa_data_img_vgg_test.h5",
  "rnn_layers": 2,
  "use_gpu": true,
  "dropout": 0.5,
  "img_seq_size": 196,
  "batch_size": 200,
  "input_json": "data/vqa_data_prepro.json",
  "emb_size": 500,
  "start_from": "",
  "seed": 1234,
  "gpuid": 2,
  "att_size": 512,
  "input_ques_h5": "data/vqa_data_prepro.h5",
  "input_img_train_h5": "data/vqa_data_img_vgg_train.h5",
  "output_size": 1000,
  "hidden_size": 1024,
  "id": "1",
  "backend": "cudnn"
}
DataLoader loading h5 question file: data/vqa_data_prepro.h5
DataLoader loading h5 image test file: data/vqa_data_img_vgg_test.h5
DataLoader loading json file: data/vqa_data_prepro.json
Accuracy on 20000 images: 43.675000%
Accuracy on 40000 images: 43.405000%
Accuracy on 60000 images: 43.403333%
Accuracy on 80000 images: 43.445000%
Accuracy on 100000 images: 43.613000%
Accuracy on 120000 images: 43.590000%
Accuracy on test set with 121512 images: 43.576766 %
abhivm@gpu:/data/abhivm/sans-pytorch$
```

Figure 21: Accuracy after 40th epoch

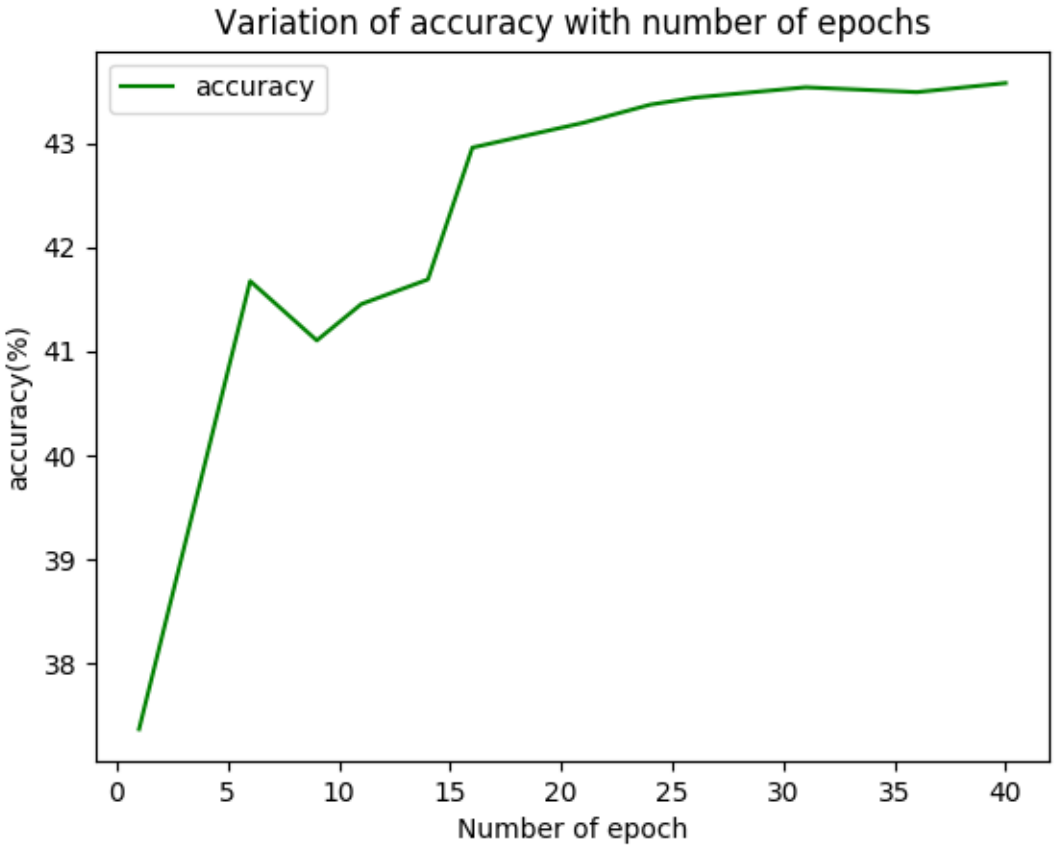

Figure 22: Variation of accuracy with number of epochs