

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]

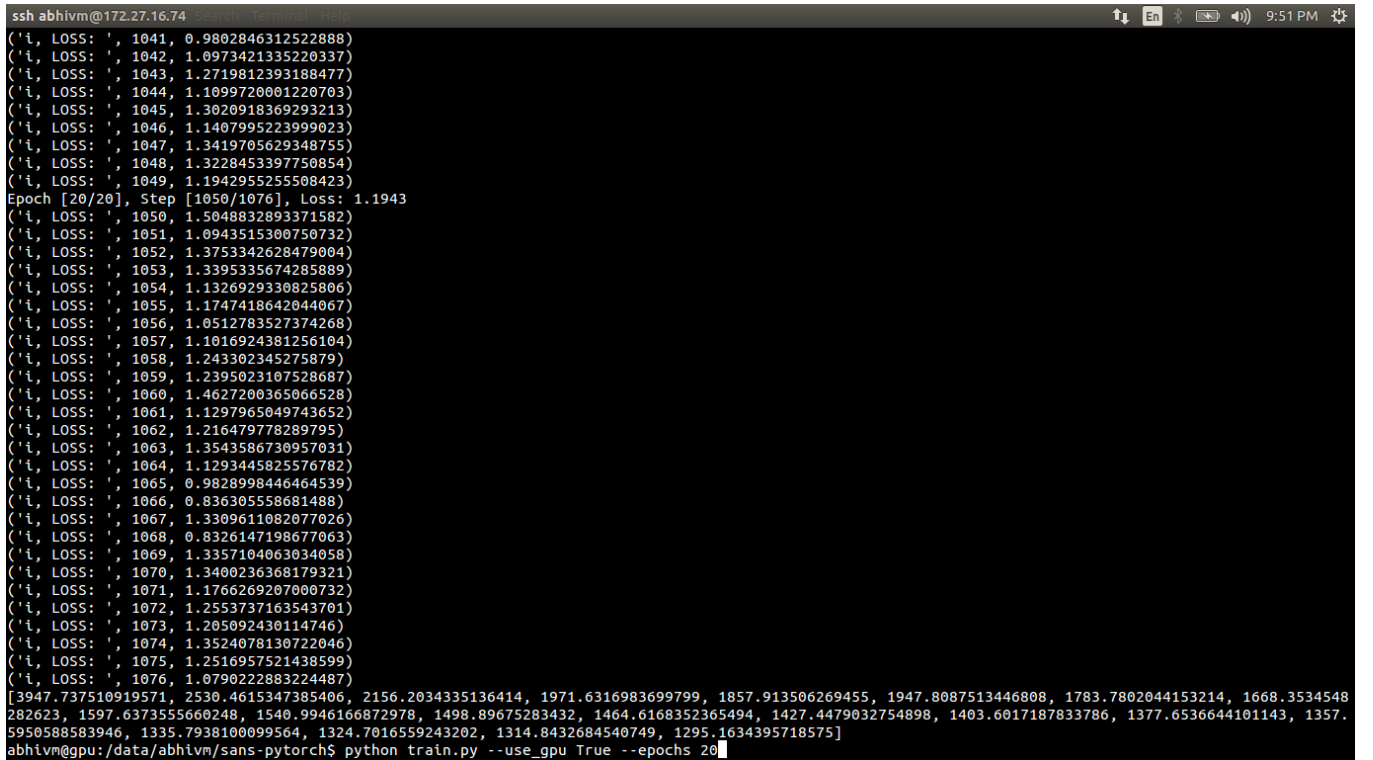


Figure 1: Train errors with LSTM of single layer

```

ssh abhivm@vyom.cc.iitk.ac.in
i: 1041 | LOSS: 0.8727 | lr: 0.000040
i: 1042 | LOSS: 0.9062 | lr: 0.000040
i: 1043 | LOSS: 1.0003 | lr: 0.000040
i: 1044 | LOSS: 0.8973 | lr: 0.000040
i: 1045 | LOSS: 0.9524 | lr: 0.000040
i: 1046 | LOSS: 0.8515 | lr: 0.000040
i: 1047 | LOSS: 1.0661 | lr: 0.000040
i: 1048 | LOSS: 0.8116 | lr: 0.000040
i: 1049 | LOSS: 0.9042 | lr: 0.000040
Epoch [20/20], Step [1050/1076], Loss: 0.9042
i: 1050 | LOSS: 1.1240 | lr: 0.000040
i: 1051 | LOSS: 0.9202 | lr: 0.000040
i: 1052 | LOSS: 1.0697 | lr: 0.000040
i: 1053 | LOSS: 0.8878 | lr: 0.000040
i: 1054 | LOSS: 0.9246 | lr: 0.000040
i: 1055 | LOSS: 1.0305 | lr: 0.000040
i: 1056 | LOSS: 0.9094 | lr: 0.000040
i: 1057 | LOSS: 0.8448 | lr: 0.000040
i: 1058 | LOSS: 0.8553 | lr: 0.000040
i: 1059 | LOSS: 1.0233 | lr: 0.000040
i: 1060 | LOSS: 1.0599 | lr: 0.000040
i: 1061 | LOSS: 0.9421 | lr: 0.000040
i: 1062 | LOSS: 0.9732 | lr: 0.000040
i: 1063 | LOSS: 0.9663 | lr: 0.000040
i: 1064 | LOSS: 0.8668 | lr: 0.000040
i: 1065 | LOSS: 0.7951 | lr: 0.000040
i: 1066 | LOSS: 0.8389 | lr: 0.000040
i: 1067 | LOSS: 0.9810 | lr: 0.000040
i: 1068 | LOSS: 0.8191 | lr: 0.000040
i: 1069 | LOSS: 1.0718 | lr: 0.000040
i: 1070 | LOSS: 0.9779 | lr: 0.000040
i: 1071 | LOSS: 0.9608 | lr: 0.000040
i: 1072 | LOSS: 0.9485 | lr: 0.000040
i: 1073 | LOSS: 0.9943 | lr: 0.000040
i: 1074 | LOSS: 0.8657 | lr: 0.000040
i: 1075 | LOSS: 1.0549 | lr: 0.000040
i: 1076 | LOSS: 0.9410 | lr: 0.000040
[4775.470651388168, 3047.539004802704, 2569.700263619423, 2454.371633887291, 2311.478813290596, 2254.9681857824326, 2112.3216720819473, 2007.6814094781
876, 1922.7496478557587, 1856.8840730190277, 1434.3337343335152, 1324.75354295969, 1267.9234648942947, 1229.401263833046, 1195.9247152209282, 1168.7838
16933632, 1144.600002527237, 1124.1185290813446, 1104.1463314890862, 1088.9114265441895]
abhivm@gpu:/data/abhivm/sans-pytorch$ python train.py --use_gpu True --batch_size 200 --epochs 20

```

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]

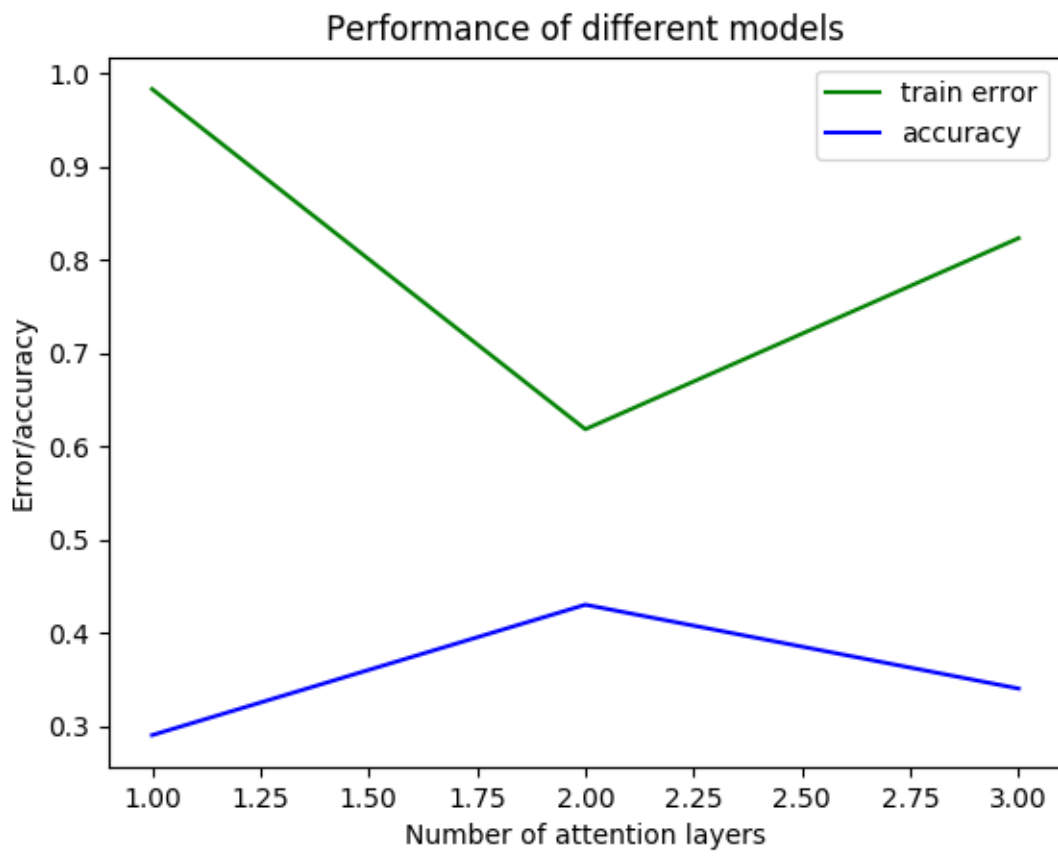


Figure 3: Performance of different models

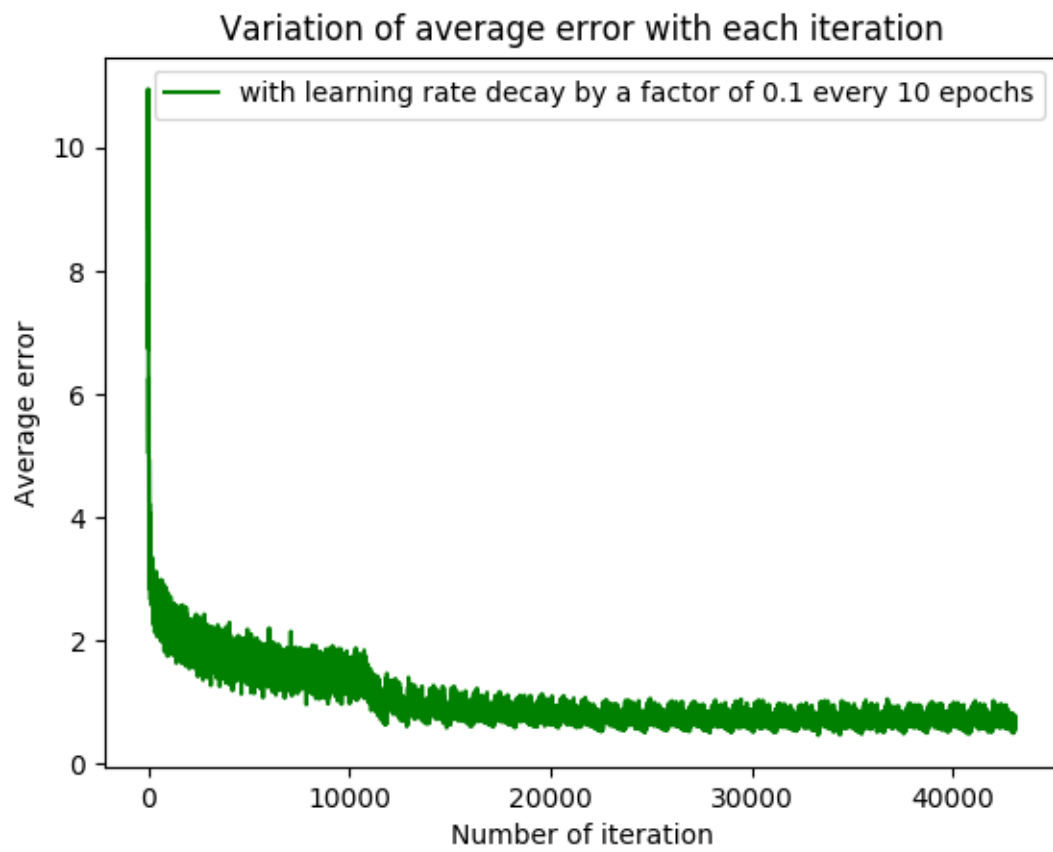


Figure 4: Variation of training error

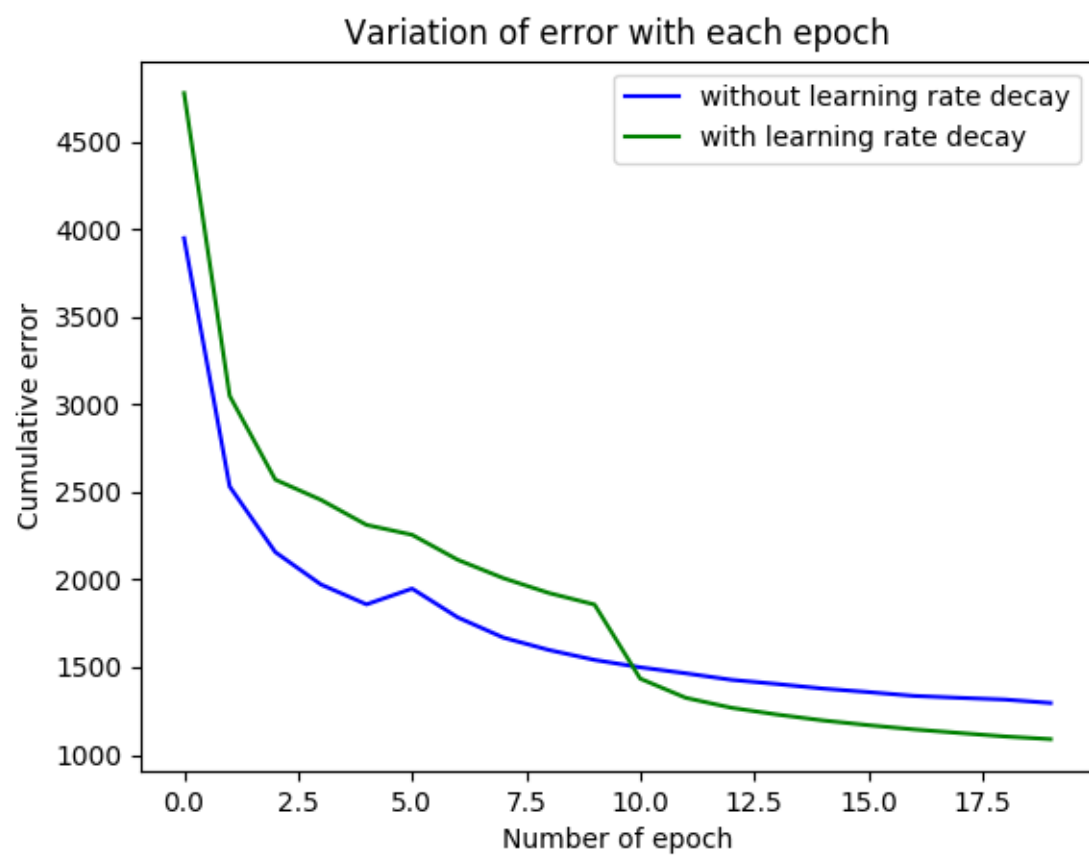


Figure 5: Variation of training error with number of epochs

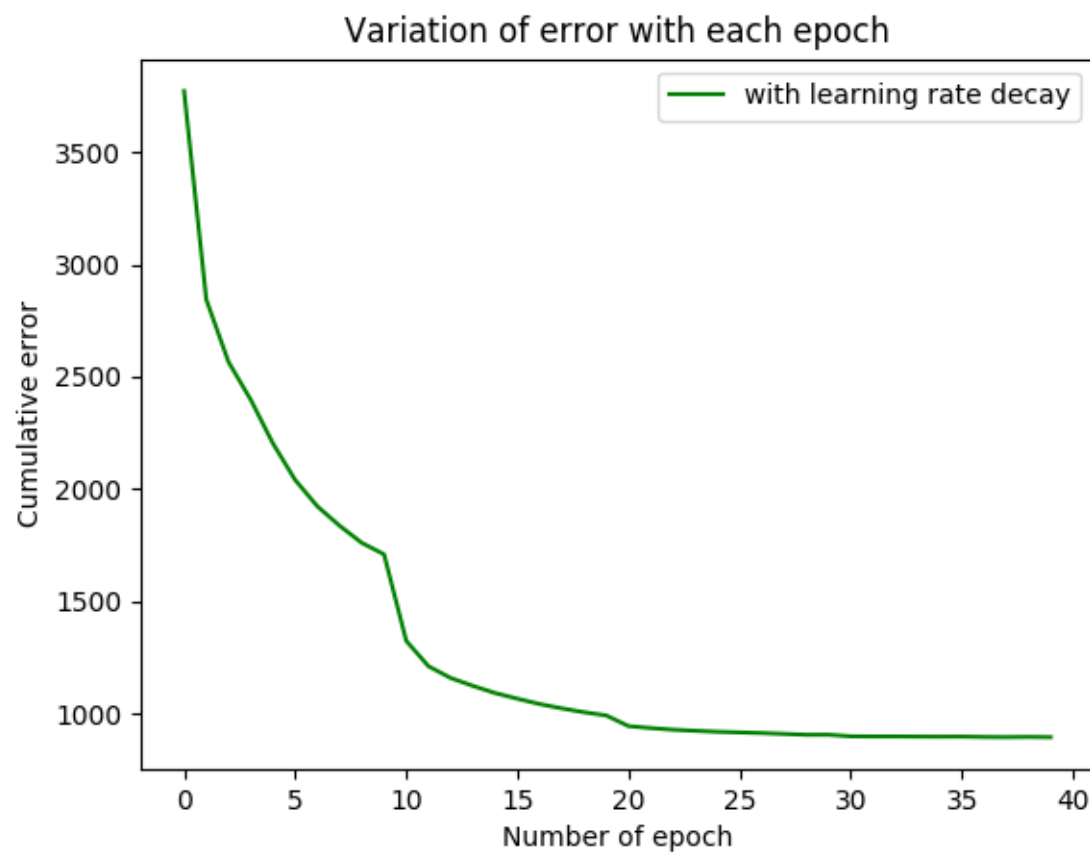


Figure 6: Variation of training error with number of epochs

```

ssh abhivm@vyom.cc.litk.ac.in
t: 1050 | LOSS: 0.4830 | lr: 0.000003
t: 1051 | LOSS: 0.5263 | lr: 0.000003
t: 1052 | LOSS: 0.5537 | lr: 0.000003
t: 1053 | LOSS: 0.5710 | lr: 0.000003
t: 1054 | LOSS: 0.5578 | lr: 0.000003
t: 1055 | LOSS: 0.5616 | lr: 0.000003
t: 1056 | LOSS: 0.5222 | lr: 0.000003
t: 1057 | LOSS: 0.5161 | lr: 0.000003
t: 1058 | LOSS: 0.5511 | lr: 0.000003
t: 1059 | LOSS: 0.5111 | lr: 0.000003
t: 1060 | LOSS: 0.6086 | lr: 0.000003
t: 1061 | LOSS: 0.5960 | lr: 0.000003
t: 1062 | LOSS: 0.5153 | lr: 0.000003
t: 1063 | LOSS: 0.6333 | lr: 0.000003
t: 1064 | LOSS: 0.5545 | lr: 0.000003
t: 1065 | LOSS: 0.6035 | lr: 0.000003
t: 1066 | LOSS: 0.5416 | lr: 0.000003
t: 1067 | LOSS: 0.6332 | lr: 0.000003
t: 1068 | LOSS: 0.4695 | lr: 0.000003
t: 1069 | LOSS: 0.5223 | lr: 0.000003
t: 1070 | LOSS: 0.5755 | lr: 0.000003
t: 1071 | LOSS: 0.5781 | lr: 0.000003
t: 1072 | LOSS: 0.5624 | lr: 0.000003
t: 1073 | LOSS: 0.5520 | lr: 0.000003
t: 1074 | LOSS: 0.6759 | lr: 0.000003
t: 1075 | LOSS: 0.5183 | lr: 0.000003
t: 1076 | LOSS: 0.5865 | lr: 0.000003
('epoch: ', 39, ' | loss_store: ', [2753.8546891212463, 2056.524384021759, 1842.0389226675034, 1706.5086417198181, 1603.839824438095, 1526.836614370346, 1463.4030282497406, 1413.0890389084816, 1371.811187326908, 1336.3571499586105, 1301.8696120381355, 1277.5583428740501, 1253.3131437301636, 1230.3100971579552, 1213.2738042473793, 934.5157096982002, 857.2580329179764, 820.8740357160568, 797.8499153256416, 777.099914252758, 760.2808124423027, 749.1864233016968, 735.338833630085, 723.8070449233055, 712.9348390102386, 704.0325638651848, 696.6753250360489, 687.0542460680008, 679.5599947571754, 671.7250701785088, 643.4591401815414, 639.1366164088249, 634.6106759309769, 632.4962058365345, 630.6912915706635, 627.9469690918922, 626.3292629420757, 625.0925199985504, 624.3324447274208, 623.005341976881])
Saving all losses to file
[2753.8546891212463, 2056.524384021759, 1842.0389226675034, 1706.5086417198181, 1603.839824438095, 1526.836614370346, 1463.4030282497406, 1413.0890389084816, 1371.811187326908, 1336.3571499586105, 1301.8696120381355, 1277.5583428740501, 1253.3131437301636, 1230.3100971579552, 1213.2738042473793, 934.5157096982002, 857.2580329179764, 820.8740357160568, 797.8499153256416, 777.099914252758, 760.2808124423027, 749.1864233016968, 735.338833630085, 723.8070449233055, 712.9348390102386, 704.0325638651848, 696.6753250360489, 687.0542460680008, 679.5599947571754, 671.7250701785088, 643.4591401815414, 639.1366164088249, 634.6106759309769, 632.4962058365345, 630.6912915706635, 627.9469690918922, 626.3292629420757, 625.0925199985504, 624.3324447274208, 623.005341976881]
abhivm@gpu:/data/abhivm/sans-pytorch$

```

Figure 7: Variation of training error with number of epochs

```

ssh abhivm@172.27.16.74
parsed input parameters:
{
  "rnn_size": 1024,
  "checkpoint_path": "train_model_lr2_ep20/",
  "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": 100,
  "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 10000 images: 38.840000%
Accuracy on 20000 images: 38.270000%
Accuracy on 30000 images: 38.066667%
Accuracy on 40000 images: 37.990000%
Accuracy on 50000 images: 38.130000%
Accuracy on 60000 images: 37.973333%
Accuracy on 70000 images: 38.085714%
Accuracy on 80000 images: 38.128750%
Accuracy on 90000 images: 38.144444%
Accuracy on 100000 images: 38.268000%
Accuracy on 110000 images: 38.281818%
Accuracy on 120000 images: 38.267500%
Accuracy on test set with 121512 images: 38.265357 %
abhivm@gpu:/data/abhivm/sans-pytorch$ python eval.py --checkpoint_path train_model_lr2_ep20/ --batch_size 100

```

Figure 8: Accuracy with learning rate=0.0002

```
ssh abhivm@172.27.16.74
parsed input parameters:
{
  "rnn_size": 1024,
  "checkpoint_path": "train_model_lr3_ep20/",
  "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": 100,
  "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 10000 images: 41.300000%
Accuracy on 20000 images: 41.295000%
Accuracy on 30000 images: 40.926667%
Accuracy on 40000 images: 40.767500%
Accuracy on 50000 images: 40.838000%
Accuracy on 60000 images: 40.775000%
Accuracy on 70000 images: 40.834286%
Accuracy on 80000 images: 40.817500%
Accuracy on 90000 images: 40.854444%
Accuracy on 100000 images: 40.986000%
Accuracy on 110000 images: 40.950909%
Accuracy on 120000 images: 40.938333%
Accuracy on test set with 121512 images: 40.935052 %
abhivm@gpu:/data/abhivm/sans-pytorch$ python eval.py --checkpoint_path train_model_lr3_ep20/ --batch_size 100
```

Figure 9: Accuracy with learning rate=0.0003

```
ssh abhivm@vyom.cc.iitk.ac.in
parsed input parameters:
{
  "rnn_size": 1024,
  "checkpoint_path": "train_model_lr4_ep20/",
  "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": 100,
  "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 10000 images: 38.790000%
Accuracy on 20000 images: 39.110000%
Accuracy on 30000 images: 39.076667%
Accuracy on 40000 images: 39.175000%
Accuracy on 50000 images: 39.316000%
Accuracy on 60000 images: 39.323333%
Accuracy on 70000 images: 39.354286%
Accuracy on 80000 images: 39.241250%
Accuracy on 90000 images: 39.296667%
Accuracy on 100000 images: 39.399000%
Accuracy on 110000 images: 39.403636%
Accuracy on 120000 images: 39.404167%
Accuracy on test set with 121512 images: 39.395286 %
abhivm@gpu:/data/abhivm/sans-pytorch$ python eval.py --checkpoint_path train_model_lr4_ep20/ --batch_size 100
```

Figure 10: Accuracy with learning rate=0.0004

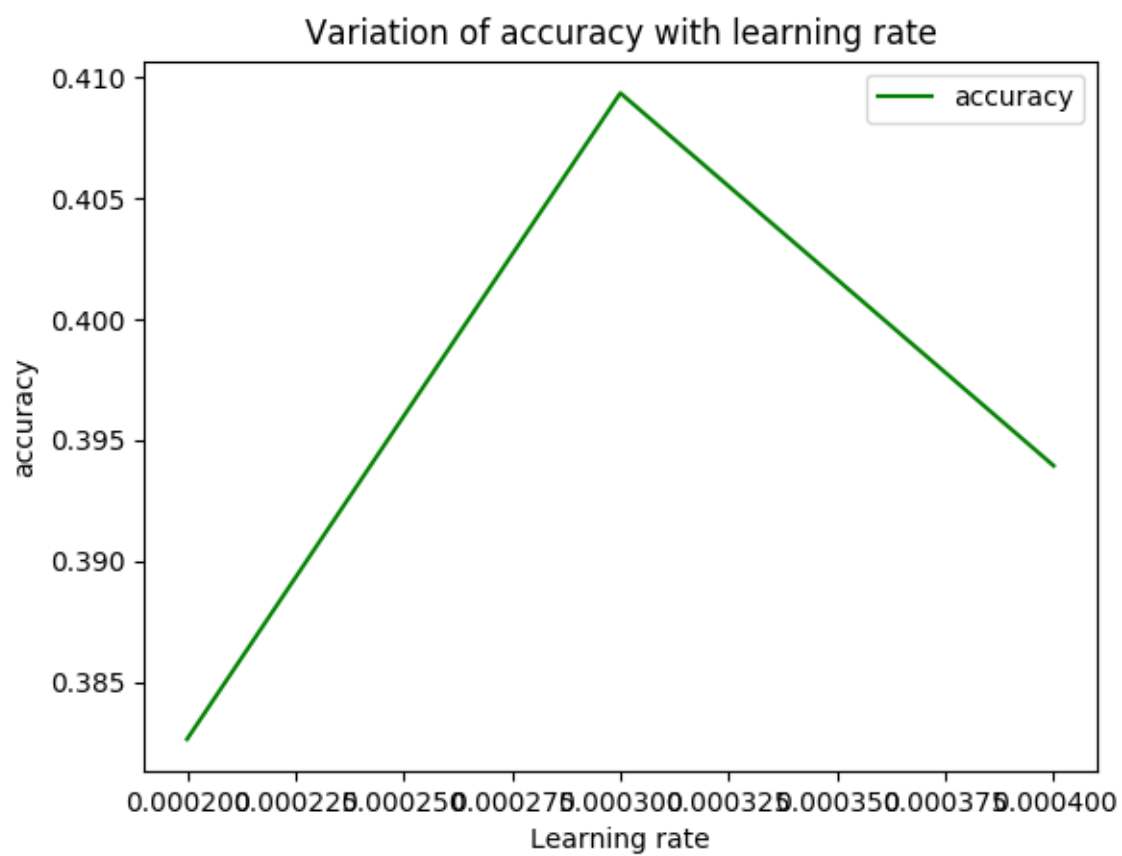


Figure 11: Variation of accuracy with learning rate

```

ssh abhivm@vyom.cc.iitk.ac.in - ssh terminal Help
return Add.apply(self, other, inplace)
File "/usr/local/lib/python2.7/dist-packages/torch/autograd/_functions/basic_ops.py", line 17, in forward
return a.add(b)
RuntimeError: cuda runtime error (2) : out of memory at /pytorch/torch/lib/THC/generic/THCStorage.cu:66
abhivm@gpu:/data/abhivm/sans-pytorch$ export CUDA_VISIBLE_DEVICES=2
abhivm@gpu:/data/abhivm/sans-pytorch$ python evaltmp.py --checkpoint_path model_lr15/ --batch_size 200
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: 37.535000%
Accuracy on 40000 images: 37.300000%
Accuracy on 60000 images: 37.298333%
Accuracy on 80000 images: 37.443750%
Accuracy on 100000 images: 37.454000%
Accuracy on 120000 images: 37.380833%
Accuracy on test set with 121512 images: 37.373264 %
abhivm@gpu:/data/abhivm/sans-pytorch$

```

Figure 12: Accuracy after 1st epoch

```

ssh abhivm@vyom.cc.iitk.ac.in - ssh terminal Help
parser.add_argument('--input_json', default='data/vqa_data_prepro.json', help='output json file')
parser.add_argument('--start_from', default='', help='path to a model checkpoint to initialize model weights from. Empty = don\'t')
"evaltmp2.py" 115L, 5596C written
abhivm@gpu:/data/abhivm/sans-pytorch$ export CUDA_VISIBLE_DEVICES=3^C
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: 40.840000%
Accuracy on 40000 images: 40.882500%
Accuracy on 60000 images: 40.945000%
Accuracy on 80000 images: 41.055000%
Accuracy on 100000 images: 41.116000%
Accuracy on 120000 images: 41.130000%
Accuracy on test set with 121512 images: 41.105405 %
abhivm@gpu:/data/abhivm/sans-pytorch$ ^C
abhivm@gpu:/data/abhivm/sans-pytorch$

```

Figure 13: Accuracy after 9th epoch

```
ssh abhivm@vyom.cc.iitk.ac.in
correct, total = 0, 0

for i, (image, question, ques_len, ans) in enumerate(test_loader):
    "evaltmp.py" 115L, 5599C written
abhivm@gpu:/data/abhivm/sans-pytorch$ python evaltmp.py --checkpoint_path model_lr15/ --batch_size 200
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: 41.230000%
Accuracy on 40000 images: 41.215000%
Accuracy on 60000 images: 41.306667%
Accuracy on 80000 images: 41.290000%
Accuracy on 100000 images: 41.410000%
Accuracy on 120000 images: 41.445000%
Accuracy on test set with 121512 images: 41.438706 %
abhivm@gpu:/data/abhivm/sans-pytorch$
```

Figure 14: Accuracy after 11th epoch

```
ssh abhivm@vyom.cc.iitk.ac.in
output = attention_model(ques_emb, img_emb)
_, prediction = torch.max(output.data, 1)
total += ans.size(0)
"evaltmp1.py" 115L, 5599C written
abhivm@gpu:/data/abhivm/sans-pytorch$ python evaltmp1.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: 41.775000%
Accuracy on 40000 images: 41.702500%
Accuracy on 60000 images: 41.706667%
Accuracy on 80000 images: 41.683750%
Accuracy on 100000 images: 41.742000%
Accuracy on 120000 images: 41.702500%
Accuracy on test set with 121512 images: 41.691356 %
abhivm@gpu:/data/abhivm/sans-pytorch$
```

Figure 15: Accuracy after 14th epoch

```
ssh abhivm@vyom.cc.iitk.ac.in
    if (params['use_gpu'] and torch.cuda.is_available()):
        image = image.cuda()
        question = question.cuda()

"evaltmp.py" 115L, 5599C written
abhivm@gpu:/data/abhivm/sans-pytorch$ python evaltmp.py --checkpoint_path model_lr15/ --batch_size 200
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: 42.955000%
Accuracy on 40000 images: 42.860000%
Accuracy on 60000 images: 42.941667%
Accuracy on 80000 images: 43.003750%
Accuracy on 100000 images: 43.079000%
Accuracy on 120000 images: 43.143333%
Accuracy on test set with 121512 images: 43.118375 %
abhivm@gpu:/data/abhivm/sans-pytorch$
```

Figure 16: Accuracy after 16th epoch

```
ssh abhivm@vyom.cc.iitk.ac.in
    output = attention_model(ques_emb, img_emb)
    _, prediction = torch.max(output.data, 1)
    total += ans.size(0)

"evaltmp.py" 115L, 5599C written
abhivm@gpu:/data/abhivm/sans-pytorch$ python evaltmp.py --checkpoint_path model_lr15/ --batch_size 200
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: 43.105000%
Accuracy on 40000 images: 43.022500%
Accuracy on 60000 images: 42.965000%
Accuracy on 80000 images: 43.036250%
Accuracy on 100000 images: 43.215000%
Accuracy on 120000 images: 43.208333%
Accuracy on test set with 121512 images: 43.195734 %
abhivm@gpu:/data/abhivm/sans-pytorch$
```

Figure 17: Accuracy after 21st epoch

```
ssh abhivm@vyom.cc.iitk.ac.in
# input json
parser.add_argument('--input_img_train_h5', default='data/vqa_data_img_vgg_train.h5', help='path to the h5file containing the train image feature')
parser.add_argument('--input_img_test_h5', default='data/vqa_data_img_vgg_test.h5', help='path to the h5file containing the test image feature')
"evaltmp.py" 115L, 5599C written
abhivm@gpu:/data/abhivm/sans-pytorch$ python evaltmp.py --checkpoint_path model_lr15/ --batch_size 200
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: 43.765000%
Accuracy on 40000 images: 43.275000%
Accuracy on 60000 images: 43.236667%
Accuracy on 80000 images: 43.330000%
Accuracy on 100000 images: 43.440000%
Accuracy on 120000 images: 43.373333%
Accuracy on test set with 121512 images: 43.369379 %
abhivm@gpu:/data/abhivm/sans-pytorch$
```

Figure 18: Accuracy after 24th epoch

```
ssh abhivm@vyom.cc.iitk.ac.in
for i, (image, question, ques_len, ans) in enumerate(test_loader):
    image = Variable(image, requires_grad=False)
    question = Variable(question, requires_grad=False)
"evaltmp1.py" 115L, 5599C written
abhivm@gpu:/data/abhivm/sans-pytorch$ python evaltmp1.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: 43.535000%
Accuracy on 40000 images: 43.292500%
Accuracy on 60000 images: 43.375000%
Accuracy on 80000 images: 43.452500%
Accuracy on 100000 images: 43.557000%
Accuracy on 120000 images: 43.544167%
Accuracy on test set with 121512 images: 43.536441 %
abhivm@gpu:/data/abhivm/sans-pytorch$
```

Figure 19: Accuracy after 31st epoch

```
ssh abhivm@vyom.cc.iiitk.ac.in
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.iiitk.ac.in
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

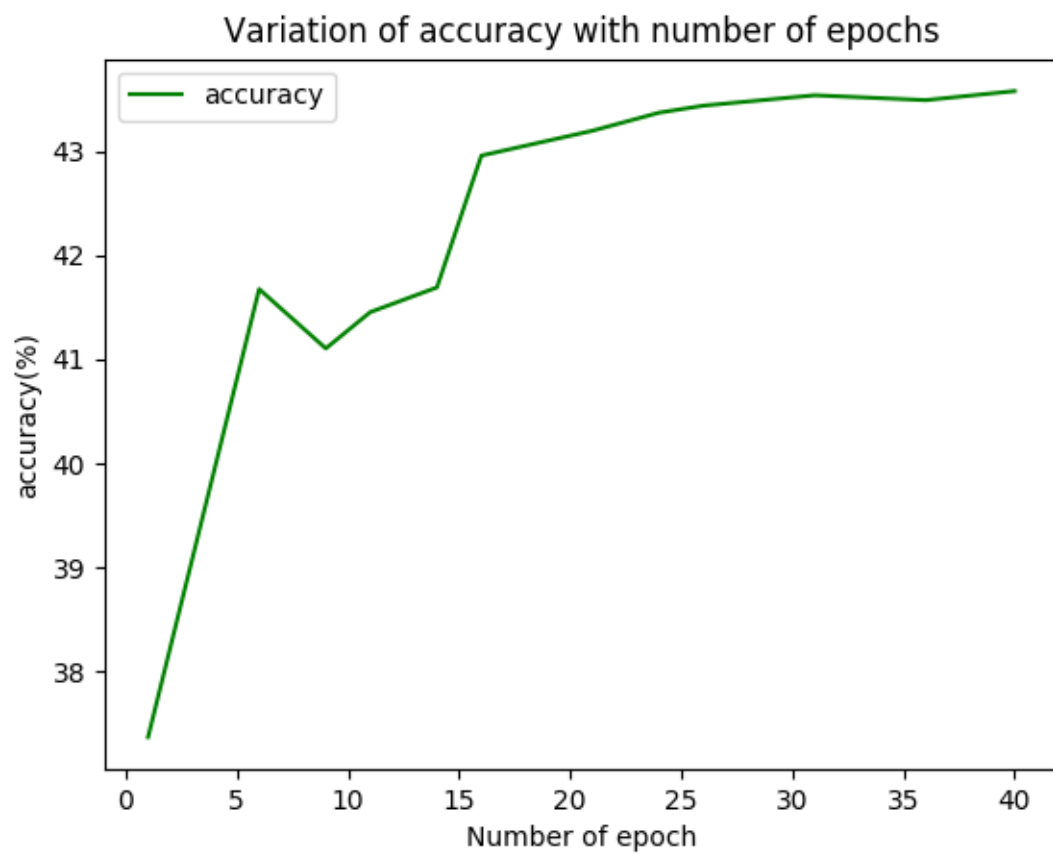


Figure 22: Variation of accuracy with number of epochs