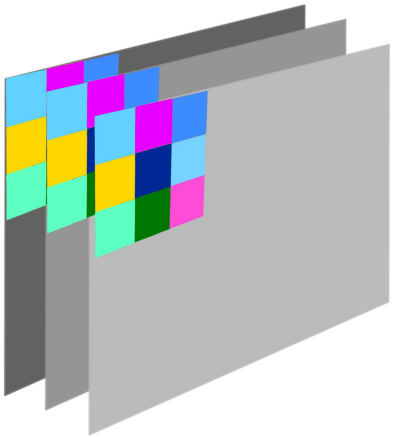
Involution: Results Presentation

Group 4

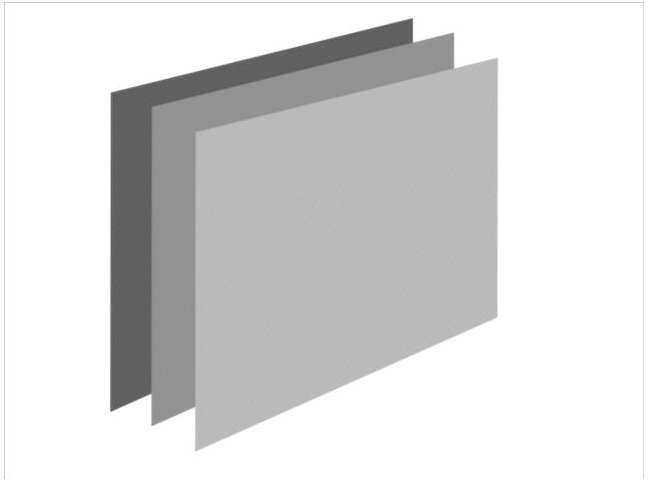
1.Harsh Chalikwar - 2021A3PS2878G
2.Aditya Kurande - 2021AAPS2485G
3.Atharv Mane - 2020A7TS0153G
4.Simran Srivastava - 2021AAPS2931G
5.Milind Kumar Prasad - 2020A7PS0130G

Two key properties of Involution

Channel-~~Specific~~ Agnostic



Spatial-~~Agnostic~~ Specific



Code

https://github.com/thatblueboy/involution

```
class ReDSNet(nn.Module):
  arch_settings = {
    26: (Bottleneck, (1, 2, 4, 1)),
    38: (Bottleneck, (2, 3, 5, 2)),
    50: (Bottleneck, (3, 4, 6, 3)),
    101: (Bottleneck, (3, 4, 23, 3)),
    152: (Bottleneck, (3, 8, 36, 3))
}


  def __init__(self,
                 depth,
               in_channels=3,
               stem_channels=64,
               base_channels=64,
               expansion = None,
               num_stages=4,
               strides=(1, 2, 2, 2),
               out_indices=(3, ),
               frozen_stages=-1,
               avg_down=False,
               zero_init_residual=True,
               is_rednet = False,
               dropout = 0.1
               ):
```

ResNet-26 —> RedNet-26
ResNet-38 —> RedNet-38
ResNet-50 —> RedNet-50
ResNet-101 —> RedNet-101
ResNet-152 —> RedNet-152

```
from models.backbones import ReDSNet
rednet26 = ReDSNet(depth = 26, is_rednet=True)
```

Classification Head

```
class Classifier(nn.Module):
    def __init__(self, input_feats, output_feats):
        super(Classifier, self).__init__()
        mid_feats = (input_feats+output_feats)//2
        self.add_module("classifier", nn.Sequential(
                nn.Linear(in_features=input_feats, out_features=mid_feats),
                nn.Linear(in_features=mid_feats, out_features=output_feats),
            )
        )
    def forward(self, x):
        logits = self.classifier(x)
        return logits
```

Pytorch Lightning

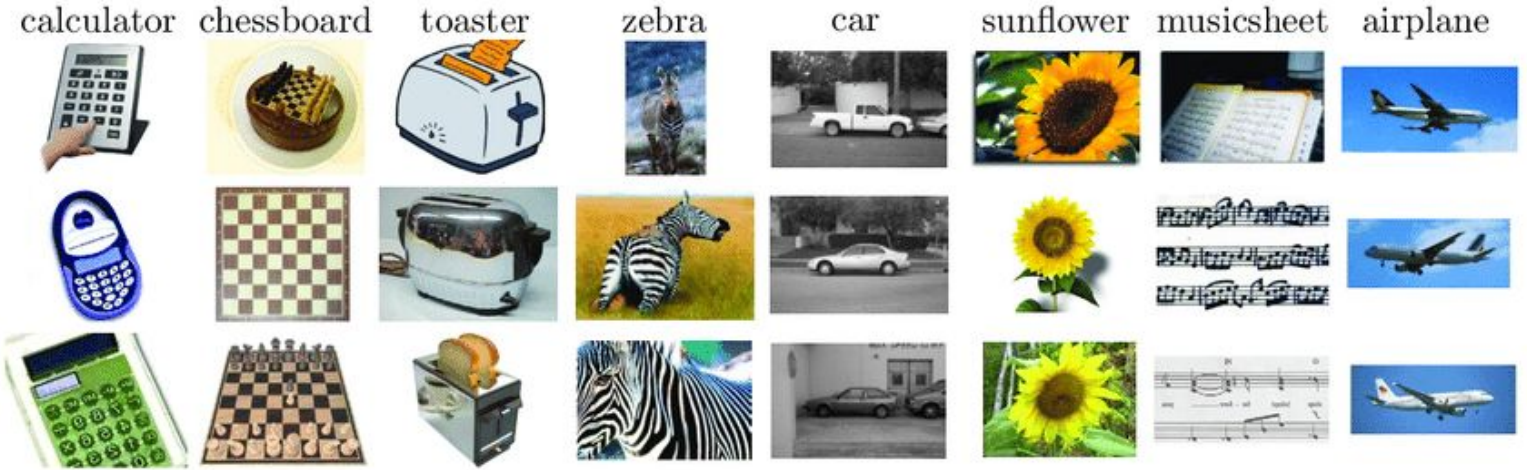
```
trainer = pl.Trainer(max_epochs=configs['max_epochs'],
                     check_val_every_n_epoch=2,
                     callbacks=[checkpoint_callback, lr_monitor],
                     logger=logger,
                     gradient_clip_val = configs['gradient_clip_val'] )
```

Wandb Logger

Name (126 visualized)	State	Notes	Use	Tag:	Crea	Runtin	Sweep	dednet	dropou	is_redr	lr_sche	lr_sche	lr_sche	num_c	optimi.	optimi.
model=<class 'models.rednet_clas…	Failed	Add notes	f20200		1d ago	4h 57m 1s	-	-	0.1	-	CosineAnr	130	0.00001	257	SGD	0.8
model=<class 'models.resnet_clas…	Finished	Add notes	f20200		2d ago	3h 45m 28	-	-	0.1	-	CosineAnr	130	0	257	SGD	0.8
imagenet_tiny-type=50-train-bs=9…	Finished	Add notes	thatblu		2d ago	53m 29s	-	-	-	true	-	130	0	200	SGD	0.012
model=<class 'models.resnet_clas…	Finished	Add notes	f20200		2d ago	2h 23m 13	-	-	0.1	-	CosineAnr	270	0	257	SGD	0.8
model=<class 'models.resnet_clas…	Finished	Add notes	f20200		2d ago	21m 32s	-	-	0.1	-	CosineAnr	270	0	257	SGD	0.0001
imagenet_tiny-type=50-dropout-c…	Finished	Add notes	thatblu		2d ago	53m 24s	-	-	-	true	-	130	0	200	SGD	0.012
imagenet_tiny-type=50-bs=96-<cla…	Finished	Add notes	thatblu		2d ago	9m 21s	-	-	-	true	-	130	0	200	SGD	0.012
model=<class 'models.resnet_clas…	Finished	Add notes	f20200		2d ago	3h 51m 22	-	-	0.1	-	CosineAnr	130	0	257	SGD	0.8
imagenet_tiny-type=50-dropout-b…	Finished	Add notes	thatblu		2d ago	1h 33m 59	-	-	-	true	-	130	0	200	SGD	0.012
model=<class 'models.resnet_clas…	Finished	Add notes	f20200		2d ago	12s	-	-	-	-	-	-	-	-	-	-
model=<class 'models.rednet_clas…	Failed	Add notes	f20200		2d ago	21m 10s	-	-	0.3	-	CosineAnr	130	0	257	SGD	0.8
model=<class 'models.rednet_clas…	Failed	Add notes	f20200		2d ago	42m 50s	-	-	0.1	-	CosineAnr	130	0	257	SGD	0.8
model=<class 'models.rednet_clas…	Finished	Add notes	f20200		2d ago	22s	-	-	0.1	-	CosineAnr	130	0	257		
Caltech256-type=121-bs=4-<class '…	Finished	Add notes	f20200		2d ago	36s	-	121	0.1	-	CosineAnr	130	0	257	SGD	0.8

1-100 of 126

Caltech 256



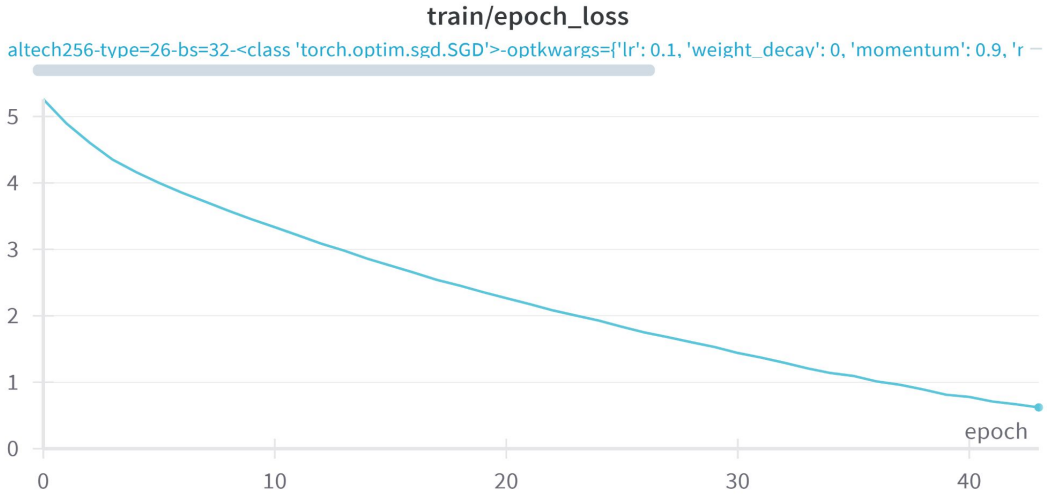
- Caltech-256 is an object recognition dataset containing 30,607 real-world images, of different sizes, spanning 257 classes (256 object classes and an additional clutter class).
- Each class is represented by at least 80 images.

Data Augmentation

- Random horizontal flip with a probability of 0.5
- Random vertical flip with a probability of 0.5
- Random rotation by up to 45 degrees
- Random affine transformation by up to 45 degrees
- Normalize the image using mean=[0.485, 0.456, 0.406] and std=[0.229, 0.224, 0.225]

Benchmarking ResNet - 26 on Caltech

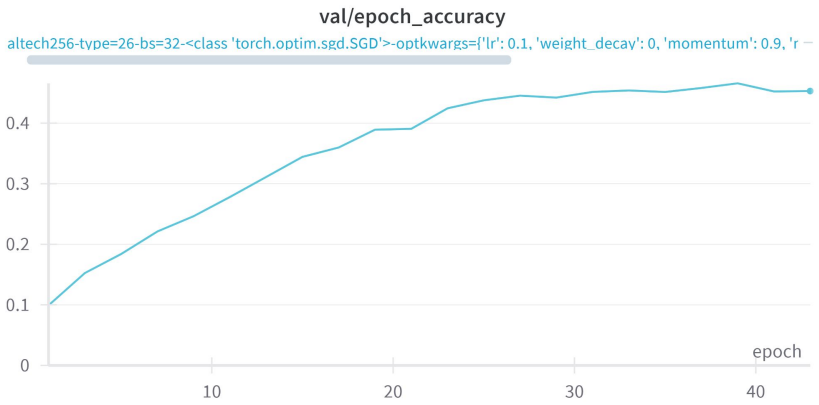
Optimiser: SGD **lr_scheduler:** CosineAnnealingLR **lr =** 0.1
weight_decay: 0.0 **grad_clip:** None

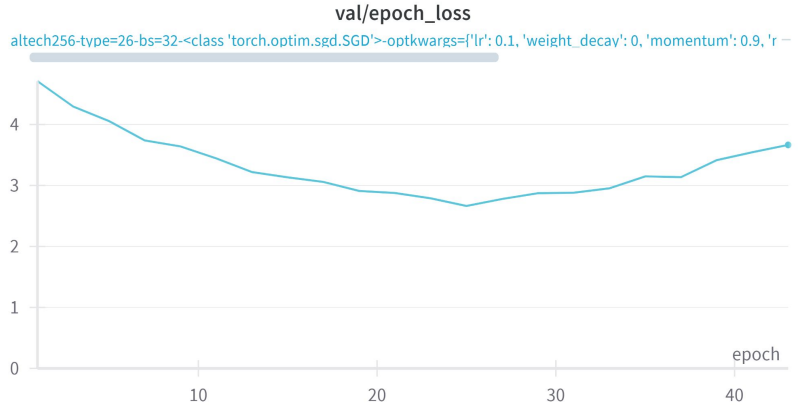


Benchmarking ResNet - 26 on Caltech

Validation loss increasing while Training loss decreases -

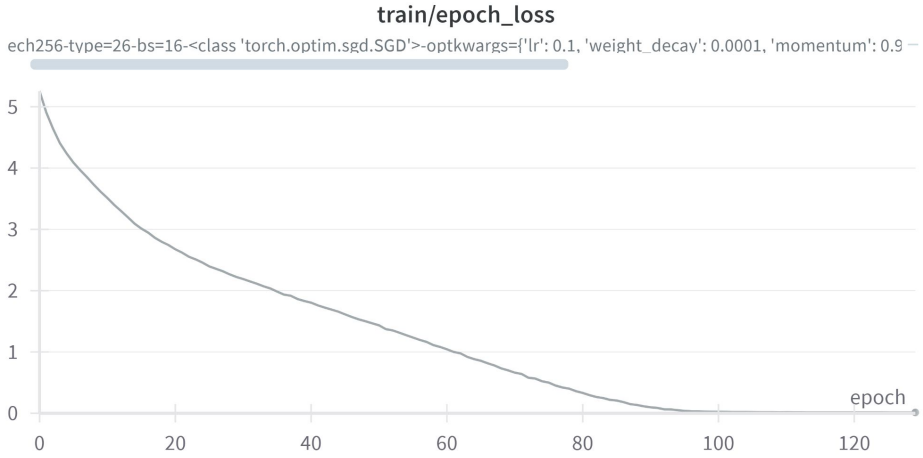
Classic Overfitting!





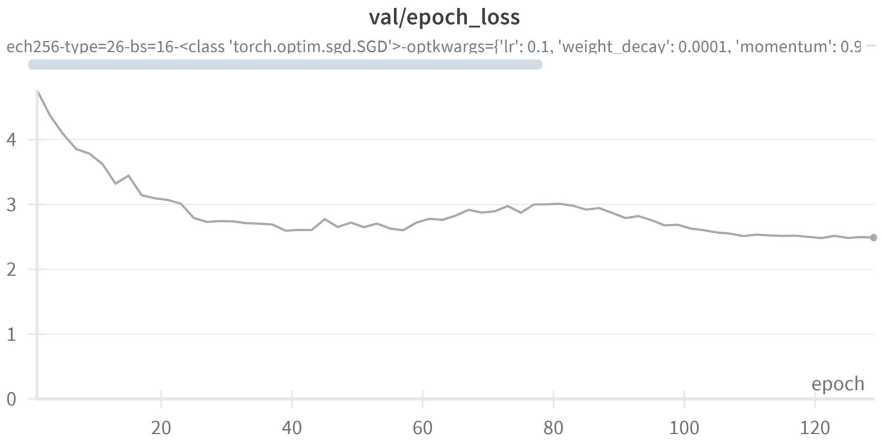
Benchmarking ResNet - 26 on Caltech

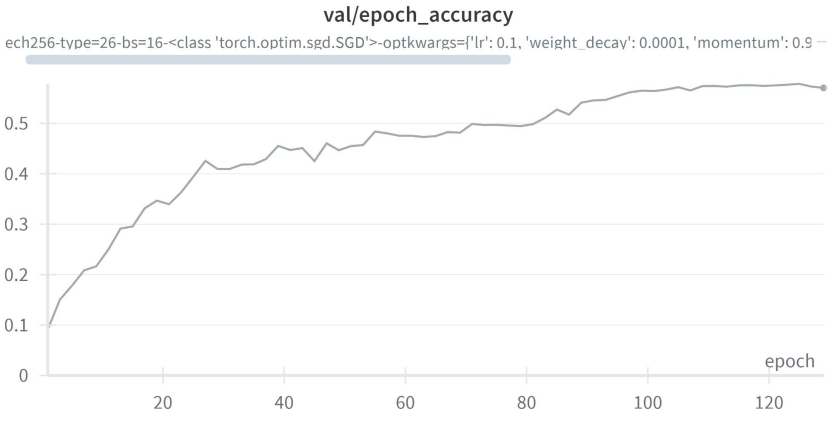
Optimiser: SGD **lr_scheduler:** CosineAnnealingLR **lr =** 0.1
weight_decay: 1e-4 **grad_clip:** None **epochs:** 130



Benchmarking ResNet - 26 on Caltech

Validation loss Decreases and Converges while Training loss converges - Overfitting Solved!

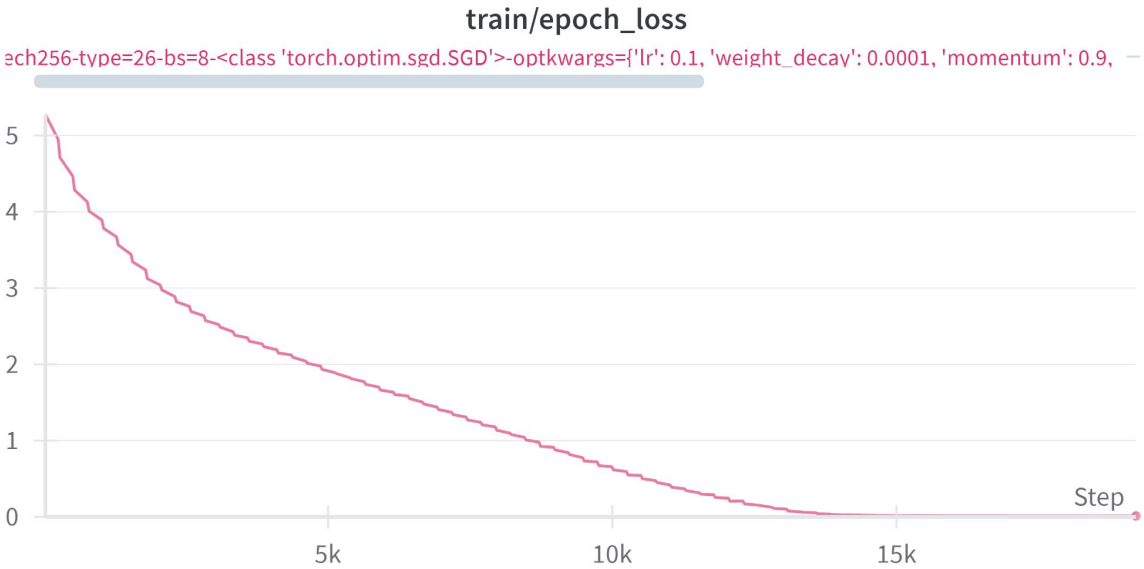




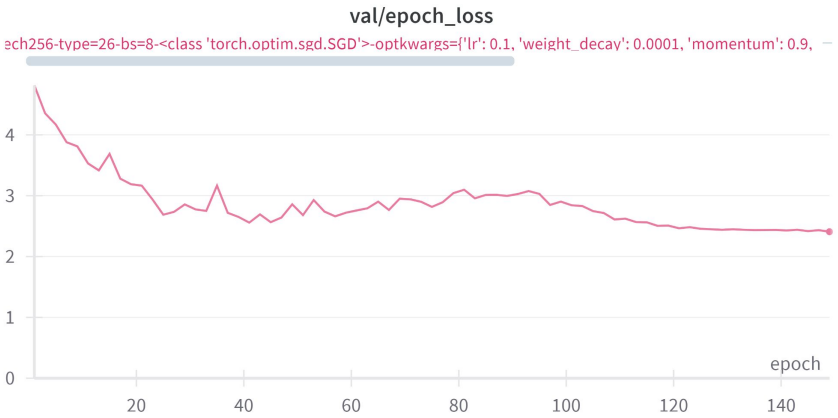
Final Accuracy for the Run: 57.03%

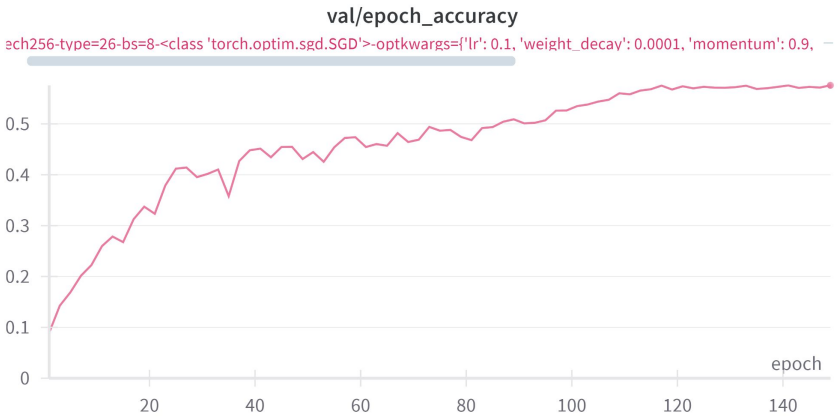
Benchmarking ResNet - 26 on Caltech: Longer Runs

Optimiser: SGD **lr_scheduler:** CosineAnnealingLR **lr =** 0.1
weight_decay: 1e-4 **grad_clip:** None **epochs:** 150



Benchmarking ResNet - 26 on Caltech





Final Accuracy for the Run: 57.59%

RedNet-50 on Caltech 256 - Another experiment to Validate NaNs - Some Internal Tensors Explode

Optimiser: SGD **lr_scheduler**: CosineAnnealingLR **lr =** 0.8 **weight_decay**: 0
grad_clip: 0

d-li14 / involution Public

Notifications | Fork 175 | Star 1.3k

Code | Issues 22 | Pull requests 2 | Actions | Projects | Security | Insights

Nan #26

New issue

Open cymdhx opened this issue on Apr 1, 2021 · 28 comments

cymdhx commented on Apr 1, 2021

when I use I meet

```
[2021-04-01 19:38:19,593]-[train.py line:232]:  === Epoch:[  0/60],step:[ 20/8275],img_size:[416],total_loss:nan|loss_ciou:nan|loss_conf:nan|loss_cls:nan|lr:0.0001
INFO:YOLOv4:  === Epoch:[  0/60],step:[ 20/8275],img_size:[416],total_loss:nan|loss_ciou:nan|loss_conf:nan|loss_cls:nan|lr:0.0001
WARNING:root:NaN or Inf found in input tensor.
WARNING:root:NaN or Inf found in input tensor.
WARNING:root:NaN or Inf found in input tensor.
WARNING:root:NaN or Inf found in input tensor.
```

Assignees

No one assigned

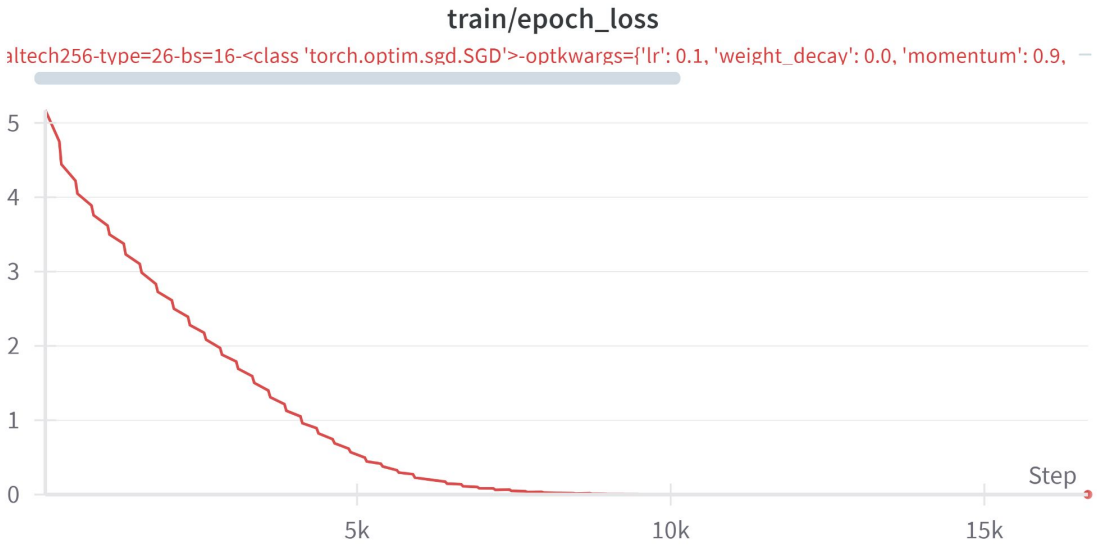
Labels

None yet

Projects

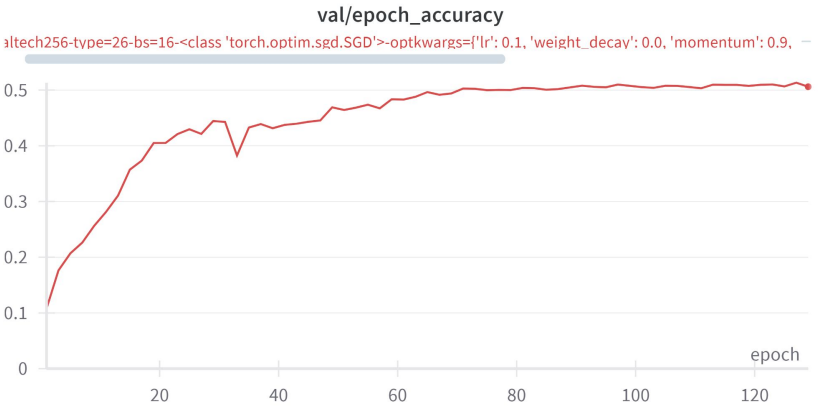
Benchmarking RedNet - 26 on Caltech

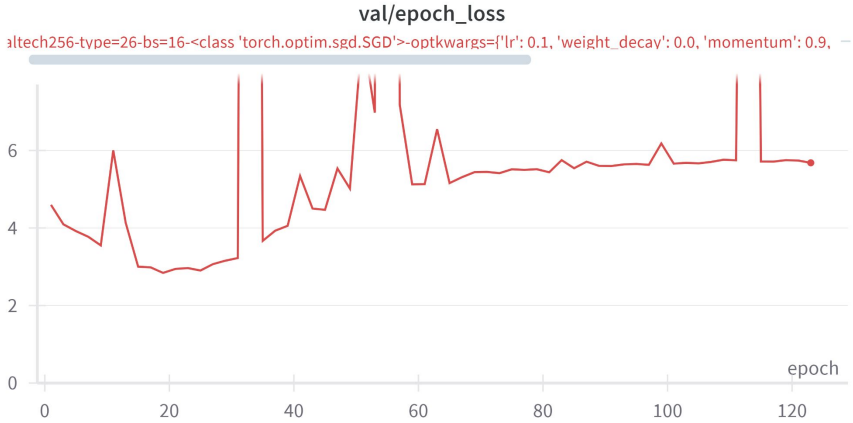
Optimiser: SGD **lr_scheduler:** CosineAnnealingLR **lr =** 0.1
weight_decay: 0.0 **grad_clip:** None **epochs:** 130



Benchmarking RedNet - 26 on Caltech

Note: Instability can be observed in validation loss. We think this is again because of **Activations exploding.** The rough shape of the curve is same as that for Resnets but with abnormalities coming in between. Also note that validation loss has increasing trend. i.e . overfitting.





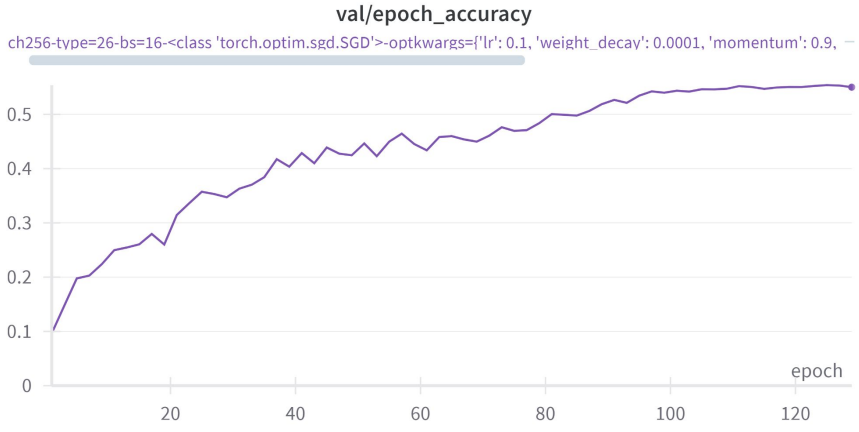
Final Accuracy for the Run: 50.6%

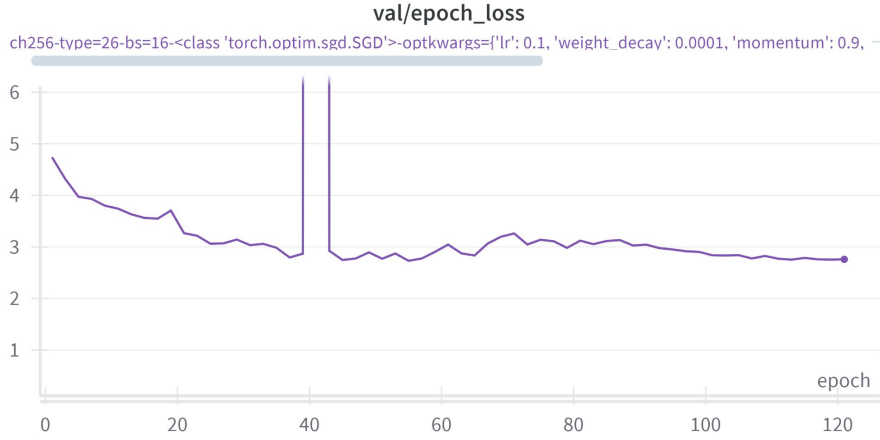
Benchmarking RedNet - 26 on Caltech

Optimiser: SGD **lr_scheduler:** CosineAnnealingLR **lr =** 0.1
weight_decay: 1e-4 **grad_clip:** None **epochs:** 130 **dropout:** 10%

Benchmarking ResNet - 26 on Caltech

Note: Irregularities in validation loss have significantly decreased.
Final Accuracy Increases.
Validation loss reduces then stabilizes. Another point to note is that activation explosion does not happen for first few validation epochs. I.E. it is learnt.





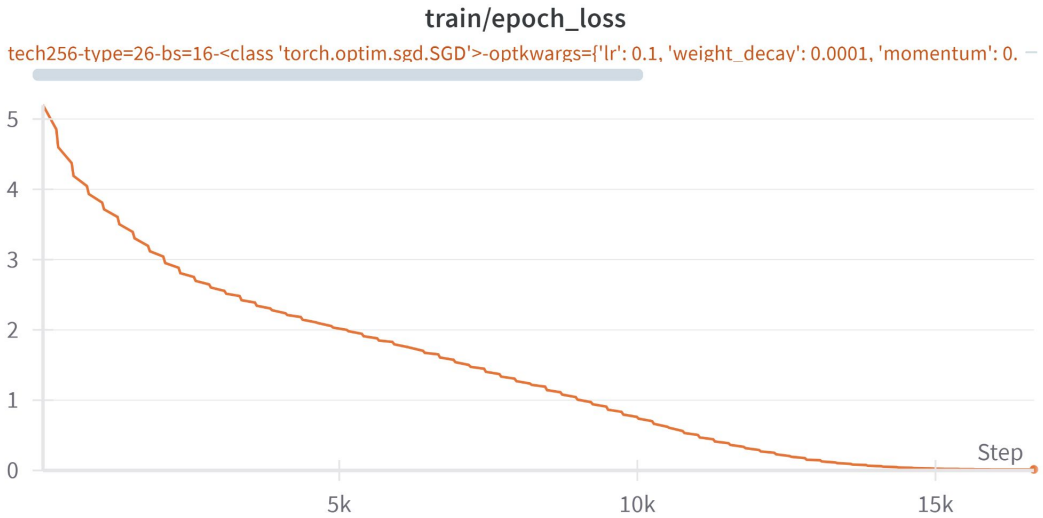
Final Accuracy for the Run: 55.00%

Benchmarking RedNet - 26 on Caltech

Optimiser: SGD **lr_scheduler:** CosineAnnealingLR **lr =** 0.1

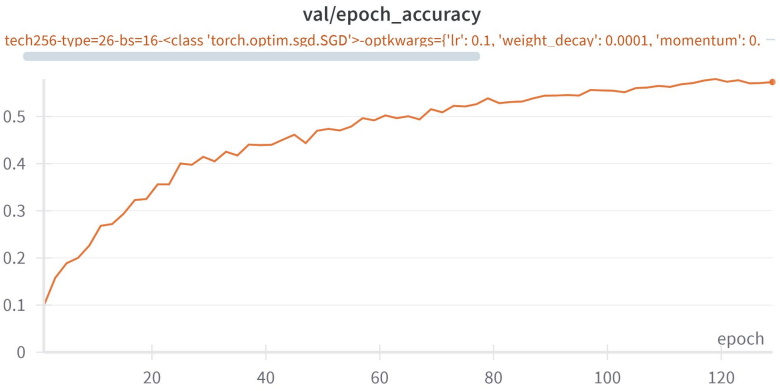
weight_decay: 1e-4 **grad_clip:** None **epochs:** 130 **dropout:** 10%

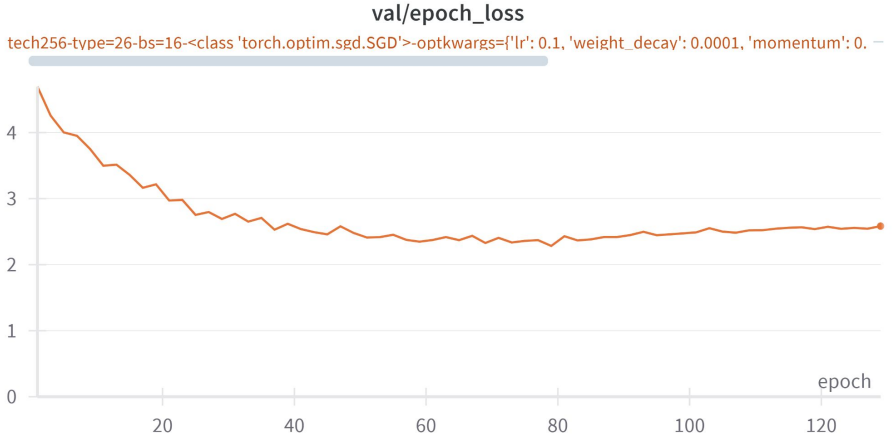
gradient clipping: 1



Benchmarking ResNet - 26 on Caltech

Note: Much stabler validation loss curve





Final Accuracy for the Run: 57.3%

Benchmarking RedNet - 26 on Caltech

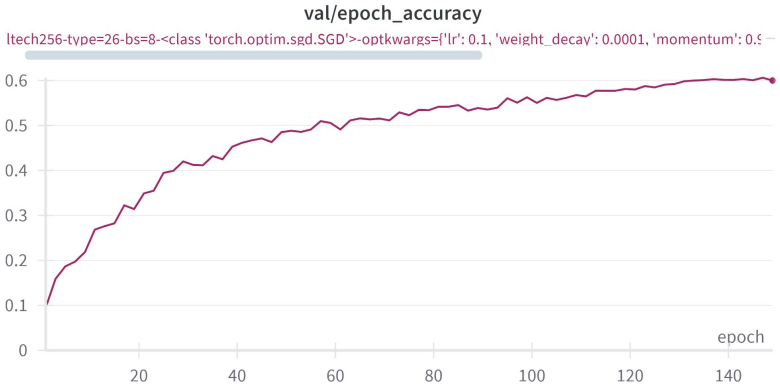
Optimiser: SGD **lr_scheduler:** CosineAnnealingLR **lr =** 0.1

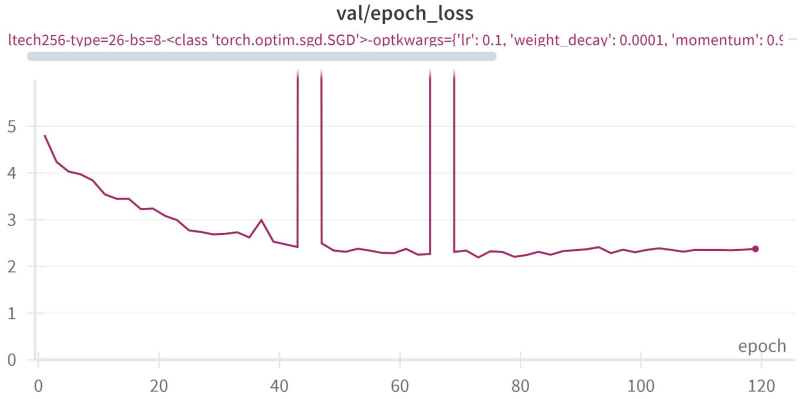
weight_decay: 1e-4 **grad_clip:** None **epochs:** 150 **dropout:** 10%

gradient clipping: 1

Benchmarking ResNet - 26 on Caltech

Note: Irregularities reappear for longer runs even with gradient clipping. This could be because of larger learning rate since the cosine annealing reduces the learning rate slower for longer runs. Nevertheless, accuracy increases.

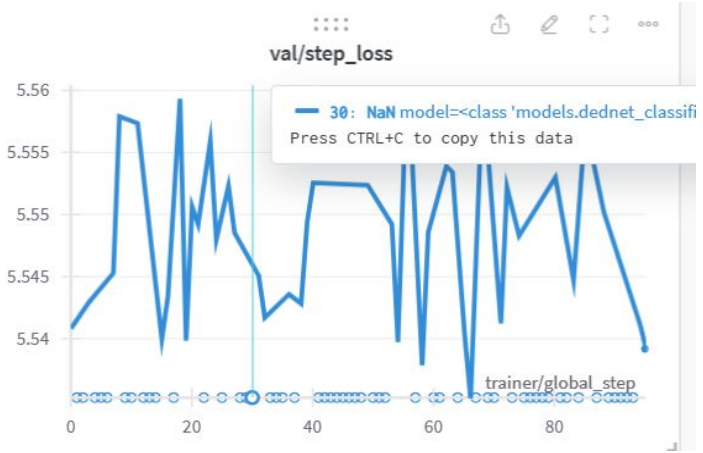


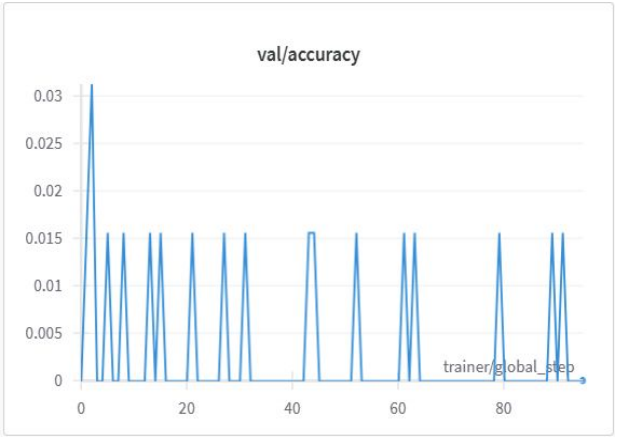


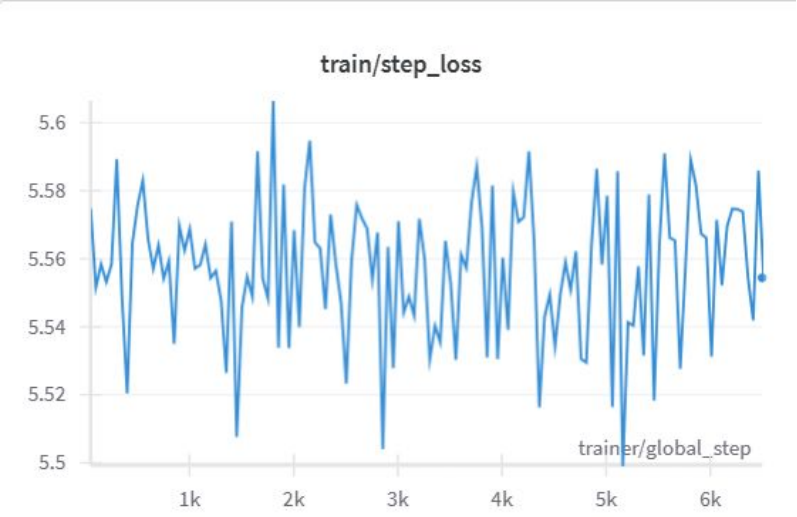
Final Accuracy for the Run: 60.05%

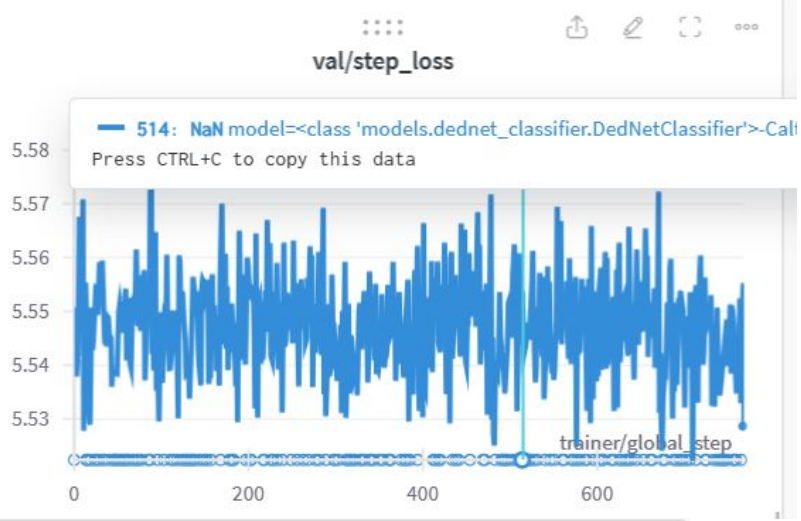
Model	No. Of Parameters	Accuracy
ResNet-26	11.7 M	57.59
RedNet-26	8 M	60.05

DEDNet: DenseNets with Involution

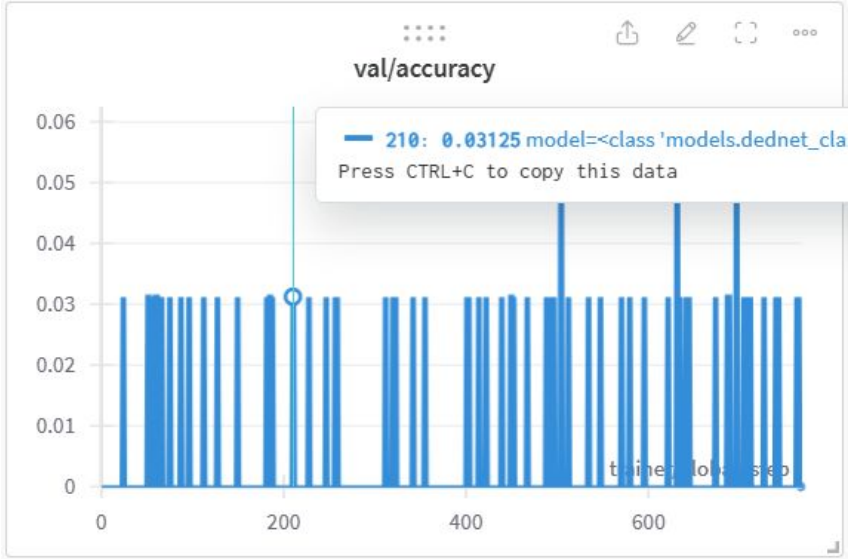








Note that the irregularities occur during “Sanity Check” runs i,e, default “kaiming initialization”. This suggests that involution in and of itself might be a method with high instability. Which gets exaggerated in densenets



Thank You