

ELEC4630 Assignment 3
HENGJI ZHAO
47184521

Question 1: Paddy Doctor: Paddy Disease Classification

This is a Kaggle competition which training deep learning modern to identify the paddy diseases.

The first step is to setup the environment for our model training. We "register" in the Kaggle website and download the corresponding `.json` file from the website then store the file in the same folder with our code. In this file, I'm about to test multiple models and test their performance on this Deep learning question, and finally find out the model with best performance.

```
In [1]: !pip install -q kaggle  
!pip install -Uq fastai  
!pip install -q timm  
  
import timm  
  
# My kaggle API  
{"username": "henryzhao1412",  
 "key": "f7c3096270e7c2e19c9c22b58bf713a9"}  
  
try: import fastkaggle  
except ModuleNotFoundError:  
    !pip install -Uq fastkaggle  
  
from fastkaggle import *  
  
import os, json  
  
# Set the Kaggle  
with open('kaggle.json') as f:  
    creds = json.load(f)  
    print(creds['username'], creds['key'])  
    os.environ['KAGGLE_USERNAME'] = creds['username']  
    os.environ['KAGGLE_KEY'] = creds['key']  
  
# Enroll the competition.  
comp = 'paddy-disease-classification'  
path = setup_comp(comp, install='fastai "timm>=0.6.2.dev0"')  
from fastai.vision.all import *  
  
# Set the seed, make sure every time our training  
# set are the same. Because we want to test multiple  
# mmodels, using the same training set is essential for  
# our model traing.  
set_seed(42)  
  
# To imporve the quality of the Deep Learning model,  
# I plan to use the image processing code which  
# Jeremy wrote in notebook 09, because I find that  
# if we do this, the error rate of the model trainging  
# will be lower.
```

```

trn_path = Path('sm1')
resize_images(path/'train_images',
              dest=trn_path,
              max_size=256, recurse=True)
files = get_image_files(trn_path)

```

henryzhao1412 f7c3096270e7c2e19c9c22b58bf713a9

In this notebook, we are planning to use two models for this competition. The four models are `resnet26d`, `convnext_small_in22k`, `convnextv2_base` and `convnextv2_large` respectively.

Firstly, we write a function to find the suitable Learning Rate for each model. This notebook is training in my own laptop and my laptop's GPU is `NVIDIA GeForce RTX 3060 Laptop GPU`. So, I need to consider the size of the input images. If the image size is too large, my laptop will cause `OutOfMemory` fault.

```

In [2]: # The variable 'arch' is the model we are going to utilized
# in this task.
def find_learning_rate(arch):

    # Form the dataloader for our model training.
    # Here we use 'aug_transforms' to create more
    # diverse training data to prevent the model from
    # overfitting.
    dls = ImageDataLoaders.from_folder(trn_path, seed=42,
                                       valid_pct=0.2,
                                       item_tfms=Resize(192, method='squish'),
                                       batch_tfms=aug_transforms(size=128, min_s
    learn = vision_learner(dls, arch, metrics=error_rate).to_fp16()

    # This is the step to find the learning rate curve.
    lrs = learn.lr_find(suggest_funcs=(valley, slide))
    print(f"Vally: {lrs.valley}, Slide: {lrs.slide}")

    recommend_learning_rate = (lrs.valley + lrs.slide) / 2

    # Return the model we formed and the dataloader
    # and the learning rate information.
    return learn, dls, recommend_learning_rate

```

Next step we are going to write the training function. The function `find_learning_rate()` can return us the formed model and in the following `train()` function, we can train it directly.

We know about the fastai training function `fine_tune`, the source code of this function:

```

def fine_tune(self:Learner, epochs, base_lr=2e-3, freeze_epochs=1,
             lr_mult=100,
             pct_start=0.3, div=5.0, **kwargs):
    "Fine tune with `freeze` for `freeze_epochs` then with `unfreeze`
    from `epochs` using discriminative LR"
    self.freeze()
    self.fit_one_cycle(freeze_epochs, slice(base_lr), pct_start=0.99,
                       **kwargs)

```

```
base_lr /= 2
self.unfreeze()
self.fit_one_cycle(epochs, slice(base_lr/lr_mult, base_lr),
pct_start=pct_start, div=div, **kwargs)
```

From this part of code, we know that the `fine_tune` function is 'composed' with two `fit_one_cycle` function. It will firstly freeze the model to do the transfer learning, that is, make the pre-trained model (like `resnet26d` in our this notebook) fixed, and only train the last few layers for some epochs. After several epochs, we unfreeze the model and train the whole model, in this way, our model can have a better performance.

In my code, I give the `train` function 2 modes, the first we can freeze the model and train several epochs, in second mode, we unfreeze and then train the whole model.

```
In [3]: def train(learn, epochs1=2, epochs2=12, learning_rate=0.001):

    learn.freeze()
    learn.fit_one_cycle(epochs1, learning_rate)

    learning_rate = learning_rate / 2

    learn.unfreeze()
    learn.fit_one_cycle(epochs2, learning_rate)

    return learn
```

Then we are going to write the function to do the test work for our model. In the predict process, we are going to apply `tta` for the prediction. `tta` will generate multiple enhanced versions of test images (like rotation, scaling, flipping), this is sort of like `aug_transform` function we write before. These generated images will be input to the model to do the prediction. Subsequently, all prediction results are fused (usually by averaging or weighted averaging) to obtain a more robust final prediction result.

Because the quantity of the input image is increased, so the time spent on `tta` function is normally more than that of `get_preds`.

```
In [4]: # Open the folder of test images.
tst_files = get_image_files(path/'test_images').sorted()
ss = pd.read_csv(path/'sample_submission.csv')
```

```
In [5]: """
This is the function for the prediction and submission.
the prediction is using the tta.
"""

def write_csv_tta(learn, dls, ss, name, describe_text):
    probs, _ = learn.tta(dl=dls.test_dl(tst_files))
    idxs = probs.argmax(dim=1)
    dls.vocab
    mapping = dict(enumerate(dls.vocab))
    results = pd.Series(idxs.numpy(), name="idxs").map(mapping)
    ss['label'] = results
```

```
ss.to_csv(name, index=False)
if not iskaggle:
    from kaggle import api
    api.competition_submit_cli(name, describe_text, comp)
```

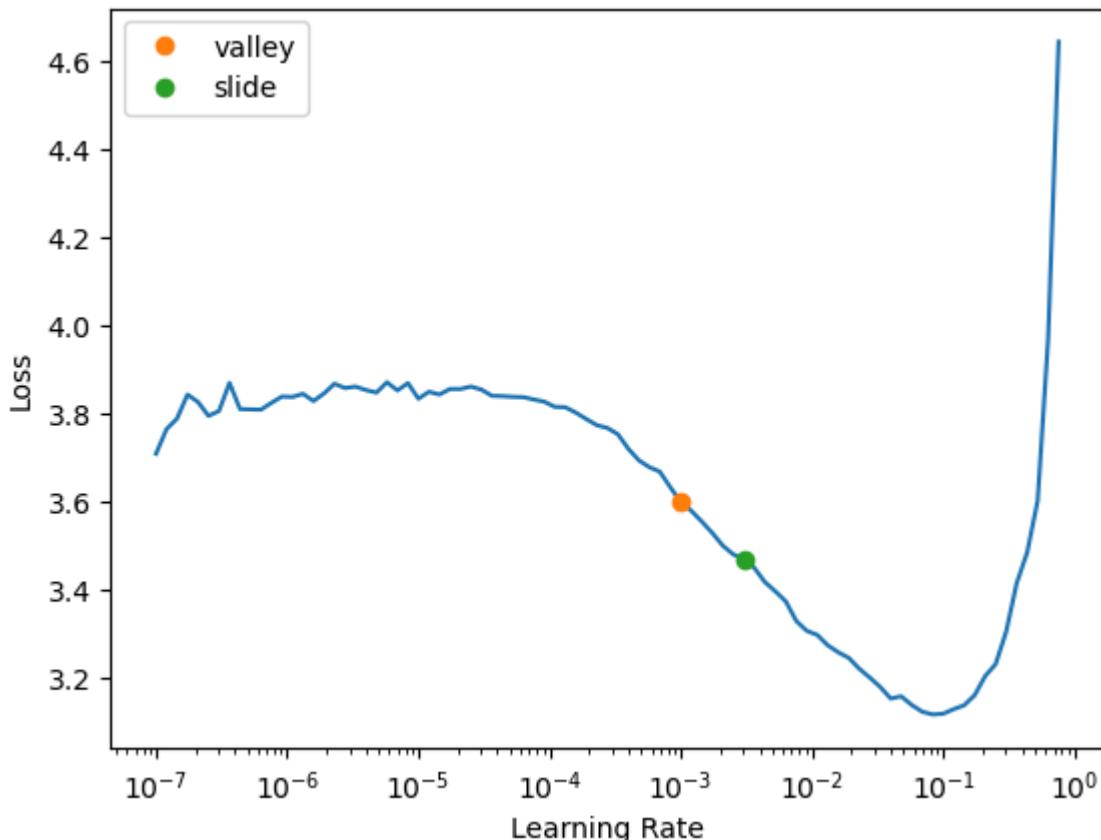
Here are the models we are going to training for the question.

```
In [6]: arch1 = 'resnet26d'
arch2 = 'convnext_small_in22k'
arch3 = 'convnextv2_base'
arch4 = 'convnextv2_large'
```

Then we are going to find the suitable learning rate for each models. Here we are going to use the function `find_learning_rate` function we just written.

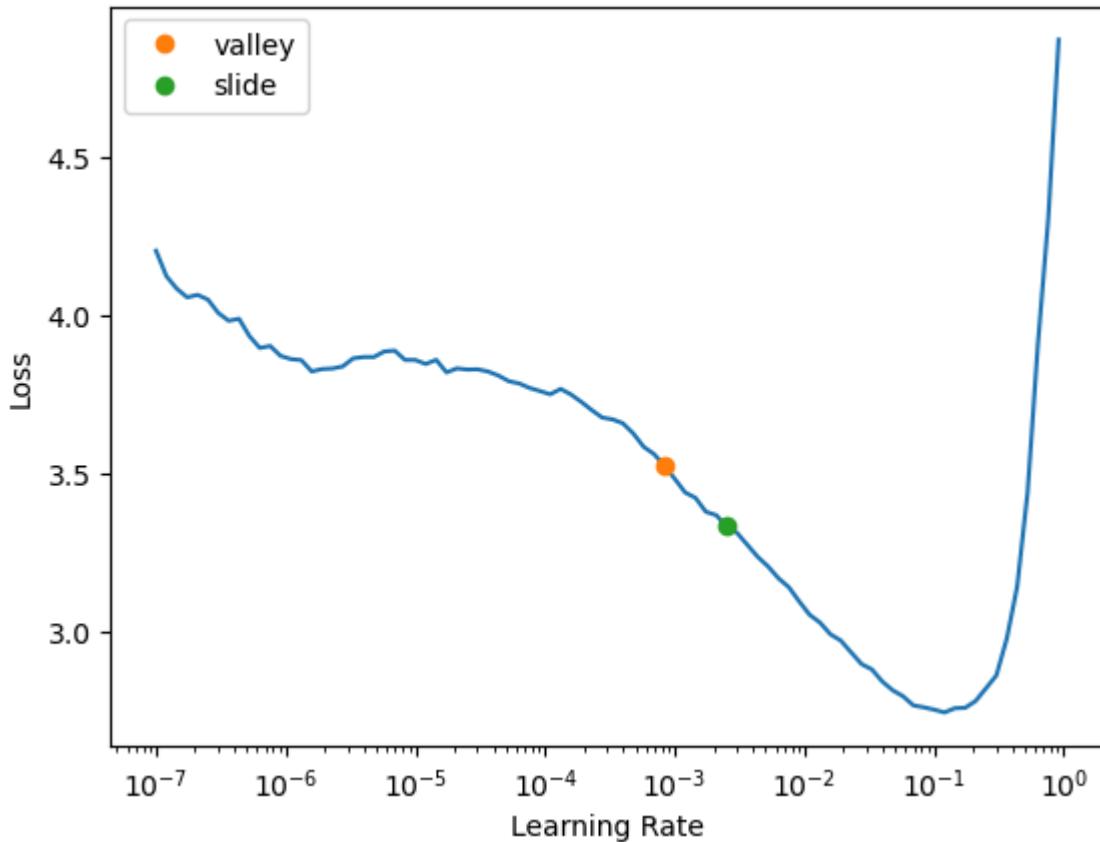
```
In [7]: learn1, dls1, lr1 = find_learning_rate(arch1)
```

```
c:\Users\henry\anaconda3\Lib\site-packages\fastai\callback\fp16.py:47: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
    self.autocast, self.learn.scaler, self.scales = autocast(dtype=dtype), GradScaler(**self.kwargs), L()
c:\Users\henry\anaconda3\Lib\site-packages\fastai\callback\fp16.py:47: FutureWarning: `torch.cuda.amp.GradScaler(args...)` is deprecated. Please use `torch.amp.GradScaler('cuda', args...)` instead.
    self.autocast, self.learn.scaler, self.scales = autocast(dtype=dtype), GradScaler(**self.kwargs), L()
Valley: 0.001000000474974513, Slide: 0.0030199517495930195
```



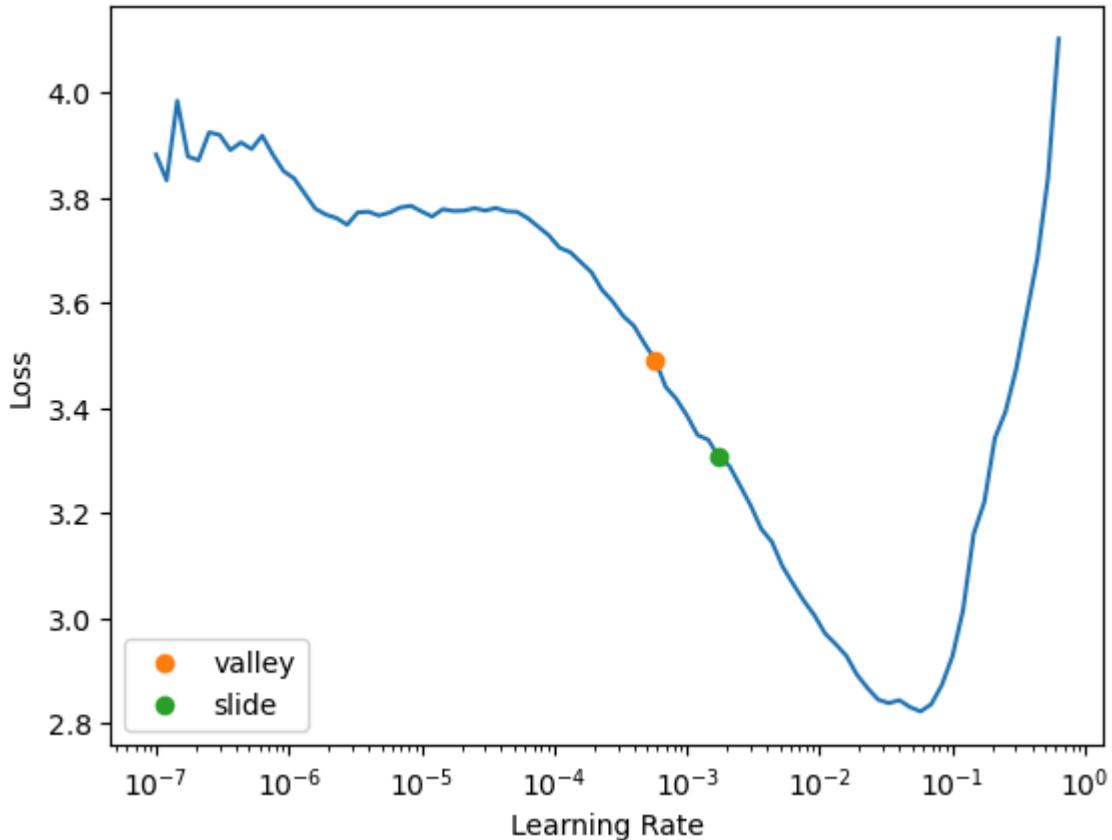
```
In [8]: learn2, dls2, lr2 = find_learning_rate(arch2)
```

```
c:\Users\henry\anaconda3\Lib\site-packages\timm\models\_factory.py:126: UserWarning: Mapping deprecated model name convnext_small_in22k to current convnext_small_fb_in22k.  
model = create_fn(  
Vally: 0.0008317637839354575, Slide: 0.002511886414140463
```



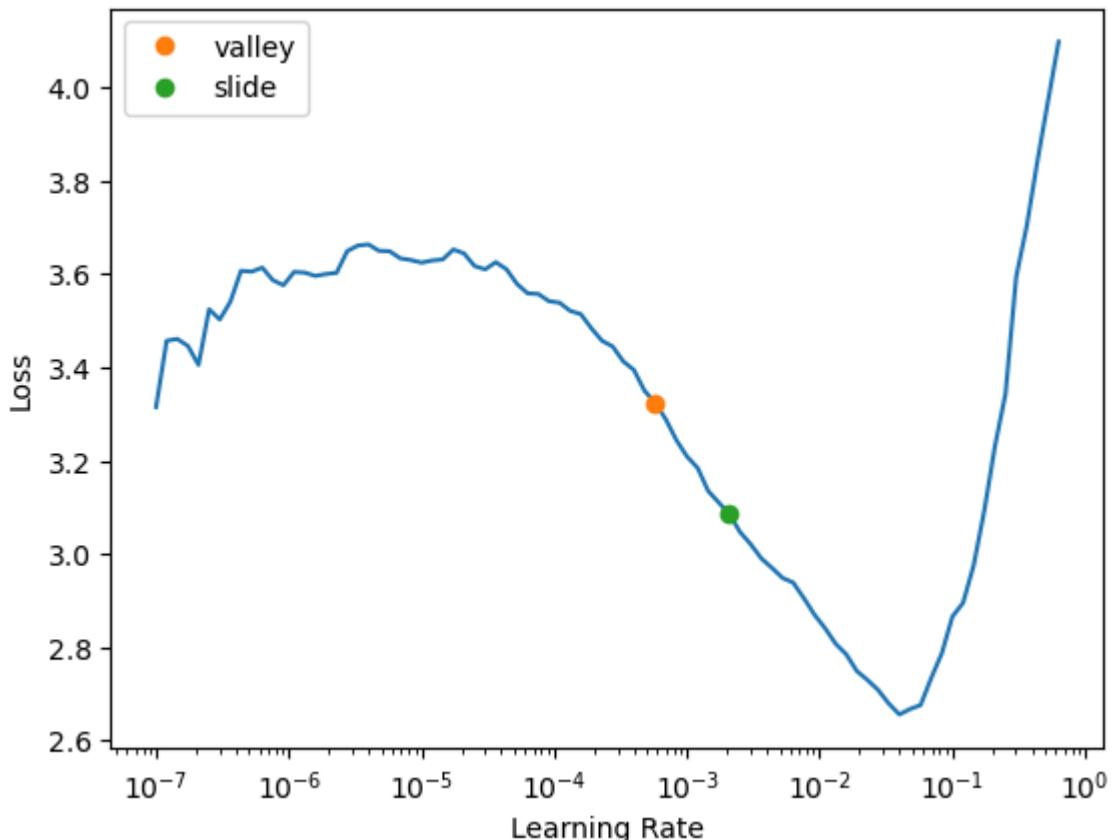
```
In [9]: learn3, dls3, lr3 = find_learning_rate(arch3)
```

```
Vally: 0.0005754399462603033, Slide: 0.001737800776027143
```



```
In [10]: learn4, dls4, lr4 = find_learning_rate(arch4)
```

Vally: 0.0005754399462603033, Slide: 0.0020892962347716093



From the figure upon, what can figure out the learning rate curve for each model. For each model, we set the learning rate for each model training between the `lr_valley` and `lr_slide`. This is a stable strategy.

In our code, we set the learning rate for the model training is the middle point between `lr_valley` and `lr_slide`.

Then we are going to train the models one by one, after we trained the model, we test the model then submit it to the Kaggle.

```
In [11]: # We first train the resnet26d model.
learn1 = train(learn1, 2, 12, lr1)
```

epoch	train_loss	valid_loss	error_rate	time
0	1.784373	1.249671	0.392119	01:29
1	1.229230	1.137965	0.367131	01:14

epoch	train_loss	valid_loss	error_rate	time
0	0.869100	0.688760	0.220087	01:17
1	0.674428	0.680706	0.200865	01:22

2	0.603385	1.032286	0.285440	01:25
3	0.476715	0.657186	0.209515	01:24
4	0.412586	0.435089	0.145603	01:24
5	0.335038	0.363596	0.109082	01:27
6	0.230447	0.297135	0.086977	01:23
7	0.176017	0.158579	0.043729	01:22
8	0.117696	0.139047	0.037482	01:22
9	0.084726	0.121639	0.033157	01:22
10	0.054755	0.108040	0.027871	01:22
11	0.048593	0.102026	0.026430	01:22

```
In [12]: name1 = "Resnet26d-training-tta"
desc1 = "This submission was trained by 'resnet26d' " \
"model with 14 epochs and tested by using TTA."
write_csv_tta(learn1, dls1, ss, name1, desc1)
```

100% |██████████| 73.9k/73.9k [00:01<00:00, 67.2kB/s]

```
In [13]: # The 'convnext_small_in22k' model.
learn2 = train(learn2, 2, 12, lr2)
```

epoch	train_loss	valid_loss	error_rate	time
0	1.273192	0.656346	0.198943	02:51
1	0.740029	0.509087	0.160980	02:52

epoch	train_loss	valid_loss	error_rate	time
0	0.576562	0.529684	0.160019	03:40
1	0.647066	0.547339	0.174916	03:40
2	0.527640	0.699336	0.221528	03:56
3	0.393418	0.469125	0.139356	03:44
4	0.291118	0.573522	0.156655	03:45
5	0.208800	0.542480	0.149928	03:53
6	0.137638	0.244405	0.075444	03:50
7	0.099062	0.129577	0.038443	04:03
8	0.060254	0.119097	0.032677	04:29
9	0.040611	0.080008	0.022105	04:31
10	0.027060	0.077419	0.020663	04:18
11	0.023060	0.078199	0.021144	04:15

```
In [14]: name2 = "convnext_small_in22k-training-tta"
desc2 = "This submission was trained by " \
"'convnext_small_in22k' " \
"model with 14 epochs and tested by using TTA."
write_csv_tta(learn2, dls2, ss, name2, desc2)
```

100% |██████████| 73.8k/73.8k [00:01<00:00, 61.8kB/s]

```
In [15]: learn3 = train(learn3, 2, 12, lr3)
```

epoch	train_loss	valid_loss	error_rate	time
0	1.350817	0.743715	0.233061	12:09
1	0.811027	0.540122	0.174435	12:11

epoch	train_loss	valid_loss	error_rate	time
0	0.657210	0.536471	0.176838	35:09
1	0.700900	0.733248	0.221528	31:27
2	0.671995	0.487433	0.147045	31:37
3	0.487290	0.514105	0.163864	30:31
4	0.356006	0.388392	0.125420	31:25
5	0.280976	0.329391	0.107160	30:37
6	0.212274	0.223168	0.063431	30:32
7	0.138857	0.190580	0.056223	31:41
8	0.079299	0.118878	0.036040	30:32
9	0.052548	0.089440	0.024027	31:01
10	0.037465	0.081201	0.024507	30:30
11	0.028025	0.078423	0.024507	30:40

```
In [16]: name3 = "convnextv2_base-training-tta"
desc3 = "This submission was trained by " \
"'convnextv2_base' " \
"model with 14 epochs and tested by using TTA."
write_csv_tta(learn3, dls3, ss, name3, desc3)
```

100% |██████████| 73.8k/73.8k [00:01<00:00, 68.0kB/s]

```
In [17]: learn4 = train(learn4, 2, 12, lr4)
```

epoch	train_loss	valid_loss	error_rate	time
0	1.169968	0.574922	0.174435	36:20
1	0.646205	0.436258	0.136953	34:22

epoch	train_loss	valid_loss	error_rate	time
0	0.543082	0.601244	0.182605	1:28:17
1	0.665919	0.577627	0.176838	1:23:32
2	0.595271	0.786382	0.222009	1:19:44
3	0.467000	0.728797	0.206151	1:17:44
4	0.335205	0.381472	0.115810	1:17:05
5	0.251640	0.331002	0.095147	1:16:18
6	0.195128	0.370173	0.111004	1:19:20
7	0.124989	0.177530	0.053820	1:16:11
8	0.086468	0.133965	0.046132	1:17:15
9	0.049165	0.104182	0.029313	1:20:16
10	0.032020	0.095268	0.024027	1:18:18
11	0.020367	0.094394	0.024027	1:16:18

```
In [18]: name4 = "convnextv2_large-training-tta"
desc4 = "This submission was trained by " \
"'convnextv2_large' " \
"model with 14 epochs and tested by using TTA."
write_csv_tta(learn4, dls4, ss, name4, desc4)
```

100% |██████████| 73.9k/73.9k [00:01<00:00, 65.1kB/s]

That's all for our training process and submission.

Then we need to export our trained model, because training these models takes a long time, we do not want to train it again when we need to use it. We should create a folder named `trained_learning_model` in the current folder.

```
In [19]: """
Create the folder and do not report fault
if the folder is existing.
"""
folder = Path('trained_models')
folder.mkdir(exist_ok=True)
```

Then we write the function for saving the model, every time we trained a model, just give the function the model we trained and the name we want to arrange is ok.

```
In [20]: save_path = Path('trained_models')
after_name = "-trained-model"

def save_model(learn, name, path=save_path):
```

```
learn.path = path
learn.export(name)
```

```
In [21]: name1 = arch1 + after_name
name2 = arch2 + after_name
name3 = arch3 + after_name
name4 = arch4 + after_name
```

```
In [22]: save_model(learn1, name1)
save_model(learn2, name2)
save_model(learn3, name3)
save_model(learn4, name4)
```

The Performance of the Trained Models.

After we submitted our models, we can review our models' performance on the Kaggle website. Figure 1.1 to 1.4 show the score of our 4 models.



Figure 1.1: The performance of the model trained by Resnet26d.



Figure 1.2: The performance of the model trained by convnext_small_in22k.



Figure 1.3: The performance of the model trained by convnextv2_base.



Figure 1.4: The performance of the model trained by convnextv2_large.

According to these models, we find that the model with the best performance is “convnext_small_in22k”, it has the highest score in “public score”, however, its “private score” is the same as the model trained by “convnextv2_base”, this means our model trained by “convnext_small_in22k” might have a problem of overfitting.

We look back to the training process of the “convnext_small_in22k” model, we find that the ‘error_rate’ reaches minimum when the epoch comes to 11 (the sequency number is 10). When the training epoch comes to 12, the error rate going up, this means the overfitting might happen.

The reason why I choose these models.

The model “Resnet26d” and “convnext_small_in22k” models are from Jeremy’s notebook chapter “o8-first-steps-road-to-the-top-part-1.ipynb” and “o9-small-models-road-to-the-top-parts-2.ipynb”. The rest models “convnextv2_base” and “convnextv2_large” are from the notebook chapter “o3-which-image-models-are-best.ipynb”. The Figure 1.5 and Figure 1.6 shows the performance of the two models.



Figure 1.5: The performance of the model “convnextv2_large”.

Inference

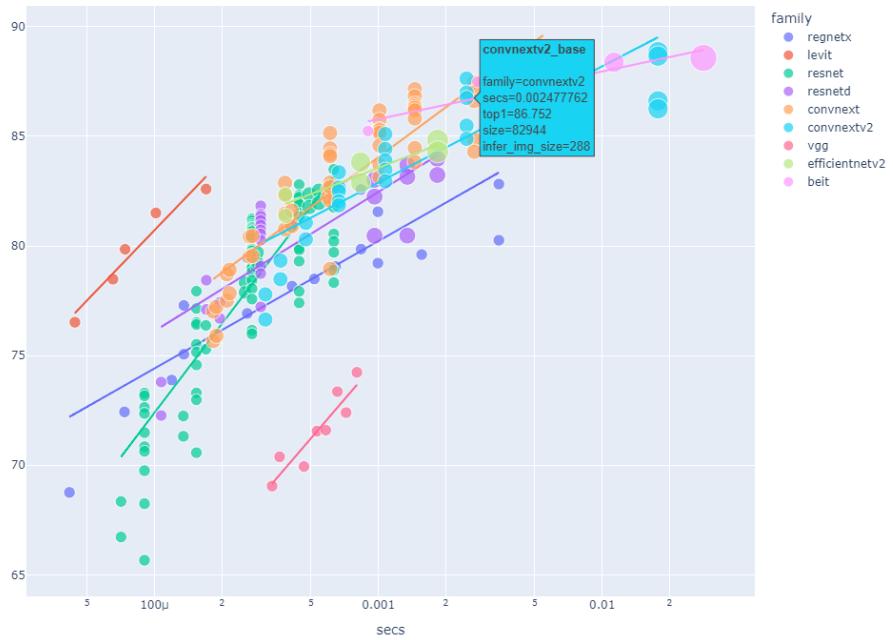


Figure 1.6: The performance of model “convnextv2_base”.

We find that the accuracy of the model compared to other models are a little bit higher, however, also the image size of models are not so large, this means our training process may not have the problem of “OutOfMemory”, which is a safe training method.

Q2: AI or REAL recognition.

Dataset prepare.

For this question, the training dataset is quite large (more than 20GB). Therefore, it is difficult for me to use my own laptop to train the model, so, I submit the training dataset to my Google Drive. And do the model training in Google Colab for a more powerful GPU.

Here I use the Nvidia A100 GPU, which is much more faster than my own laptop. To use the Google Colab to train my model, I firstly submitted the `zip` file contains training images to my Google Drive.

Firstly, access to my own Google Drive, where stored my training dataset.

```
In [ ]: from google.colab import drive  
drive.mount('/content/drive')
```

Mounted at /content/drive

We first list the content in our current directory, we find there are no training dataset. This is because currently we haven't unzip it.

I tried to unzip the file and store the unzipped images in my Google Drive, however, I find that training speed is much more slower than what I expected, therefore I went to check the reference here:

- [Why-my-google-colab-GPU-is-slow \(<https://amit-manbansh01.medium.com/have-you-ever-encountered-that-your-keras-image-classification-model-is-training-very-slow-during-962f1c1bb95f>\)](https://amit-manbansh01.medium.com/have-you-ever-encountered-that-your-keras-image-classification-model-is-training-very-slow-during-962f1c1bb95f) [1]

In this article, the author talked why sometimes when we use Google Colab GPU training model is slow, this is because if we store the images in Google Drive, every time when we need to utilize the image, the Colab will access the Drive, the time consumed for one image is around 40ms, it seems not very long. However, our training dataset contains more than 20000 images, the sum of the time will be extremely long.

The solution given by the author is unzip the dataset every time when we need it. This is because every time when we run the Google Colab notebook, it will create a virtual environment for the code, the environment will be closed when we turn off the notebook. However the speed for access the virtual environment is much more faster than Google Drive (nearly 40 times faster). That's why I choose to unzip it here. Compare to the time lost in accessing the Google Drive, the time for unzip process can be ignored.

```
In [ ]: !ls
```

drive sample_data

We then unzip the dataset, the system will print every unzipped image here, this will make the output bar give a lot of information (about 5000 lines for each), these information is not important, and print the image information will waste a long time. Then we add `-q` in our

```
In [ ]: # Unzip the training dataset.
!unzip -q '/content/drive/My Drive/archive.zip' -d '/content/training'
```

```
In [ ]: # Unzip the testing dataset.
!unzip -q '/content/drive/My Drive/archive_2.zip' -d '/content/testing'
```

Set the environment.

Here we find the unzipped folders in our current dictionary. There are `testing` and `training` folder.

```
In [ ]: !ls
```

```
drive sample_data testing training
```

Then we upload the `kaggle.json` file and the `sample.sav` to Colab notebook.

```
In [ ]: from google.colab import files
files.upload()
```

No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

Saving `kaggle.json` to `kaggle.json`

```
Out[6]: {'kaggle.json': b'{"username":"henryzhao1412","key":"bb10c2026d0a342cdc80df3d8b1c3143"}'}
```

```
In [ ]: files.upload()
```

```
g,0\n53.jpg,1\n54.jpg,0\n55.jpg,1\n56.jpg,0\n57.jpg,0\n58.jpg,1\n59.jpg,1\n60.jpg,0\n61.jpg,0\n62.jpg,0\n63.jpg,0\n64.jpg,0\n65.jpg,1\n66.jpg,0\n67.jpg,1\n68.jpg,1\n69.jpg,0\n70.jpg,1\n71.jpg,0\n72.jpg,1\n73.jpg,1\n74.jpg,0\n75.jpg,1\n76.jpg,0\n77.jpg,1\n78.jpg,0\n79.jpg,1\n80.jpg,0\n81.jpg,0\n82.jpg,1\n83.jpg,0\n84.jpg,0\n85.jpg,0\n86.jpg,0\n87.jpg,0\n88.jpg,1\n89.jpg,0\n90.jpg,0\n91.jpg,1\n92.jpg,0\n93.jpg,0\n94.jpg,0\n95.jpg,1\n96.jpg,1\n97.jpg,1\n98.jpg,0\n99.jpg,0\n100.jpg,0\n101.jpg,0\n102.jpg,1\n103.jpg,0\n104.jpg,0\n105.jpg,0\n106.jpg,0\n107.jpg,1\n108.jpg,0\n109.jpg,1\n110.jpg,1\n111.jpg,0\n112.jpg,1\n113.jpg,0\n114.jpg,0\n115.jpg,0\n116.jpg,1\n117.jpg,0\n118.jpg,0\n119.jpg,0\n120.jpg,0\n121.jpg,0\n122.jpg,1\n123.jpg,1\n124.jpg,0\n125.jpg,0\n126.jpg,1\n127.jpg,1\n128.jpg,0\n129.jpg,0\n130.jpg,1\n131.jpg,0\n132.jpg,0\n133.jpg,1\n134.jpg,0\n135.jpg,1\n136.jpg,0\n137.jpg,1\n138.jpg,1\n139.jpg,0\n140.jpg,0\n141.jpg,1\n142.jpg,0\n143.jpg,0\n144.jpg,1\n145.jpg,0\n146.jpg,1\n147.jpg,0\n148.jpg,1\n149.jpg,1\n150.jpg,0\n151.jpg,0\n152.jpg,0\n153.jpg,1\n154.jpg,1\n155.jpg,0\n156.jpg,0\n157.jpg,0\n158.jpg,1\n159.jpg,1\n160.jpg,1\n161.jpg,0\n162.jpg,0\n163.jpg,1\n164.jpg,0\n165.jpg,0\n166.jpg,1\n167.jpg,0\n168.jpg,1\n169.jpg,1\n170.jpg,1\n171.jpg,0\n172.jpg,0\n173.jpg,0\n174.jpg,1\n175.jpg,1\n176.jpg,1\n177.jpg,1\n178.jpg,0\n179.jpg,1\n
```

Here we can check whether the two files uploaded successfully.

In []: !ls

```
drive  kaggle.json  sample.csv  sample_data  testing  training
```

Here we set the Kaggle API.

```
In [ ]: import os, json
```

```
with open('kaggle.json') as f:  
    creds = json.load(f)  
    print(creds['username'], creds['key'])  
    os.environ['KAGGLE_USERNAME'] = creds['username']  
    os.environ['KAGGLE_KEY'] = creds['key']
```

henryzhao1412 bb10c2026d0a342cdc80df3d8b1c3143

Import the necessary libraries.

```
In [ ]: from fastcore.all import *
!pip install -Uqq fastbook
from fastbook import *
from fastai.vision.all import *
from fastai.vision.widgets import *
```

```
eta 0:00:00          719.8/719.8 kB 13.1 MB/s
eta 0:00:00          363.4/363.4 MB 3.0 MB/s
eta 0:00:00          13.8/13.8 MB 127.4 MB/s
eta 0:00:00          24.6/24.6 MB 100.1 MB/s
eta 0:00:00          883.7/883.7 kB 59.9 MB/s
eta 0:00:00          664.8/664.8 MB 1.7 MB/s
eta 0:00:00          211.5/211.5 MB 11.5 MB/s
eta 0:00:00          56.3/56.3 MB 42.7 MB/s e
ta 0:00:00           127.9/127.9 MB 8.7 MB/s
eta 0:00:00          207.5/207.5 MB 12.6 MB/s
eta 0:00:00          21.1/21.1 MB 107.7 MB/s
eta 0:00:00          1.6/1.6 MB 73.8 MB/s eta
0:00:00
```

```
In [ ]: import os
iskaggle = os.environ.get('KAGGLE_KERNEL_RUN_TYPE', '')

if iskaggle:
    !pip install -Uqq fastai

from time import sleep
import os
from glob import glob
```

Enroll in the competition, after our training finished, we need to submit the trained files.

```
In [ ]: try: import fastkaggle
except ModuleNotFoundError:
    !pip install -Uq fastkaggle

!pip install -q kaggle
!pip install -Uq fastai
!pip install -q timm

from fastkaggle import *

import timm

comp = 'hack-rush-deep-fake-detection'
path = setup_comp(comp, install='fastai "timm>=0.6.2.dev0"')

----- 235.3/235.3 kB 8.0 MB/s
eta 0:00:00
----- 78.2/78.2 kB 6.9 MB/s et
a 0:00:00
----- 42.6/42.6 kB 3.4 MB/s et
a 0:00:00
----- 1.2/1.2 MB 35.7 MB/s eta
0:00:00
Downloading hack-rush-deep-fake-detection.zip to /content
100%|██████████| 12.3k/12.3k [00:00<00:00, 16.5MB/s]
```

Images Pre-Processing

```
In [ ]: trn_path = Path('sml')
sto_path = Path('training')

# Resize the images to 400x400 maximum.
resize_images(sto_path, dest=trn_path, max_size=400, recurse=True)

/usr/local/lib/python3.11/dist-packages/PIL/Image.py:1043: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (90671520 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (98806617 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (161087488 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (98058240 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn()
```

Check whether our pre-processing is successful.

```
In [ ]: Path('sml').ls()
```

```
Out[14]: (#2) [Path('sml/ai'), Path('sml/real')]
```

```
In [ ]: Path('training').ls()
```

```
Out[15]: (#2) [Path('training/ai'), Path('training/real')]
```

Check whether we can access all the images. If the images can be opened, we need to delete it.

```
In [ ]: # Delete the failed images, which will influence our model training.
failed = verify_images(get_image_files(trn_path))
failed.map(Path.unlink)
len(failed)
```

```
Out[16]: 0
```

Here is the code for us to build our model, I prepared 2 methods of the input images, the first is the `squish` method and the second is `zero-padding` method. These two methods are from Jeremy's notebook:

- [08-first-steps-road-to-the-top-part-1.ipynb](https://github.com/fastai/course22/blob/master/08-first-steps-road-to-the-top-part-1.ipynb)
[\(https://github.com/fastai/course22/blob/master/08-first-steps-road-to-the-top-part-1.ipynb\)](https://github.com/fastai/course22/blob/master/08-first-steps-road-to-the-top-part-1.ipynb)

- [1.ipynb](#) [2]
- [09-small-models-road-to-the-top-part-2.ipynb](#)
(<https://github.com/fastai/course22/blob/master/09-small-models-road-to-the-top-part-2.ipynb>) [3]
- [10-scaling-up-road-to-the-top-part-3.ipynb](#)
(<https://github.com/fastai/course22/blob/master/10-scaling-up-road-to-the-top-part-3.ipynb>) [4]

These methods he utilized to do the paddy-disease-classification competition, which can also be utilized to do the AI & REAL classification.

```
In [ ]: def find_learning_rate_squish(arch):

    dls = ImageDataLoaders.from_folder(trn_path, seed=42, valid_pct=0.2,
                                         item_tfms=Resize(224, method='squish'),
                                         batch_tfms=aug_transforms(size=192))

    learn = vision_learner(dls, arch, metrics=error_rate)
    lrs = learn.lr_find(suggest_funcs=(valley, slide))
    print(f"Valley: {lrs.valley}; Slide: {lrs.slide}")
    lr = (lrs.valley + lrs.slide)/2

    return learn, dls, lr

def find_learning_rate_zero(arch):

    dls = ImageDataLoaders.from_folder(trn_path, seed=42, valid_pct=0.2,
                                         item_tfms=Resize((256,192), method='squish'),
                                         batch_tfms=aug_transforms(size=(192,144)))

    learn = vision_learner(dls, arch, metrics=error_rate)
    lrs = learn.lr_find(suggest_funcs=(valley, slide))
    print(f"Valley: {lrs.valley}; Slide: {lrs.slide}")
    lr = (lrs.valley + lrs.slide)/2

    return learn, dls, lr
```

DataBlock method was from the Assignment 2 task. The reference is here:

- [00-is-it-a-bird-creating-a-model-from-your-own-data.ipynb](#)
(<https://github.com/fastai/course22/blob/master/00-is-it-a-bird-creating-a-model-from-your-own-data.ipynb>) [5]

Here I regard AI and REAL as two different categories. And use this method to determine it.

```
In [ ]: def datablock_method(arch):

    dls = DataBlock(
        blocks=(ImageBlock, CategoryBlock),
        get_items=get_image_files,
        splitter=RandomSplitter(valid_pct=0.2, seed=42),
        get_y=parent_label,
        item_tfms=[Resize(192, method='squish')])
    .dataloaders(trn_path)

    learn = vision_learner(dls, arch, metrics=error_rate)
    lrs = learn.lr_find(suggest_funcs=(valley, slide))
    print(f"Valley: {lrs.valley}; Slide: {lrs.slide}")
    lr = (lrs.valley + lrs.slide)/2

    return learn, dls, lr
```

In this assignment, we are going to try multiple models, they are `resnet26d` , `convnext_small_in22k` , `convnextvs_base` , `convnextv2_large` . And we are also about to test some models brought by fastai `resnet18` , `resnet34` , `resnet50` and `resnet101` .

Combined with the previous mentioned three methods, we will train 24 models and test their performance by using the Kaggle score.

```
In [ ]: """
The model, 'resnet18', 'resnet34', 'resnet50'
and 'resnet101' we can use it directly in fastai
environment, therefore we should write the rest models
in the form of a variable in our notebook.
"""

arch1 = 'resnet26d'
arch2 = 'convnext_small_in22k'
arch3 = 'convnextv2_base'
arch4 = 'convnextv2_large'
```

According to the rules of this competition, if the image is identified as `ai` , we write it to `0` otherwise we write `1` .

When we finish our training, we can use this function to write the data in the `.csv` file and submit it to the Kaggle website.

```
In [ ]: ss = pd.read_csv(Path('sample.csv'))  
  
test_images = Path('testing/teamspace/studios/this_studio/attacked_im  
test_files = [test_images/fname for fname in ss['filename']]  
  
def write_to_submission(dls, learn, ss, name, describe_text):  
  
    test_dl = dls.test_dl(test_files)  
    probs, _ = learn.tta(dl=test_dl)  
    preds = probs.argmax(dim=1).numpy()  
    ss['class'] = preds  
  
    ss.to_csv(name, index=False)  
  
    if not iskaggle:  
        from kaggle import api  
        api.competition_submit_cli(name, describe_text, comp)
```

Here we start our training process, we firstly try the -[Resnet18](#), [Resnet34](#), [Resnet50](#) and [Resnet101](#) (<https://fastai1.fast.ai/vision.models.html>) [5] model. These models are included in fastai libraries so we can use them directly.

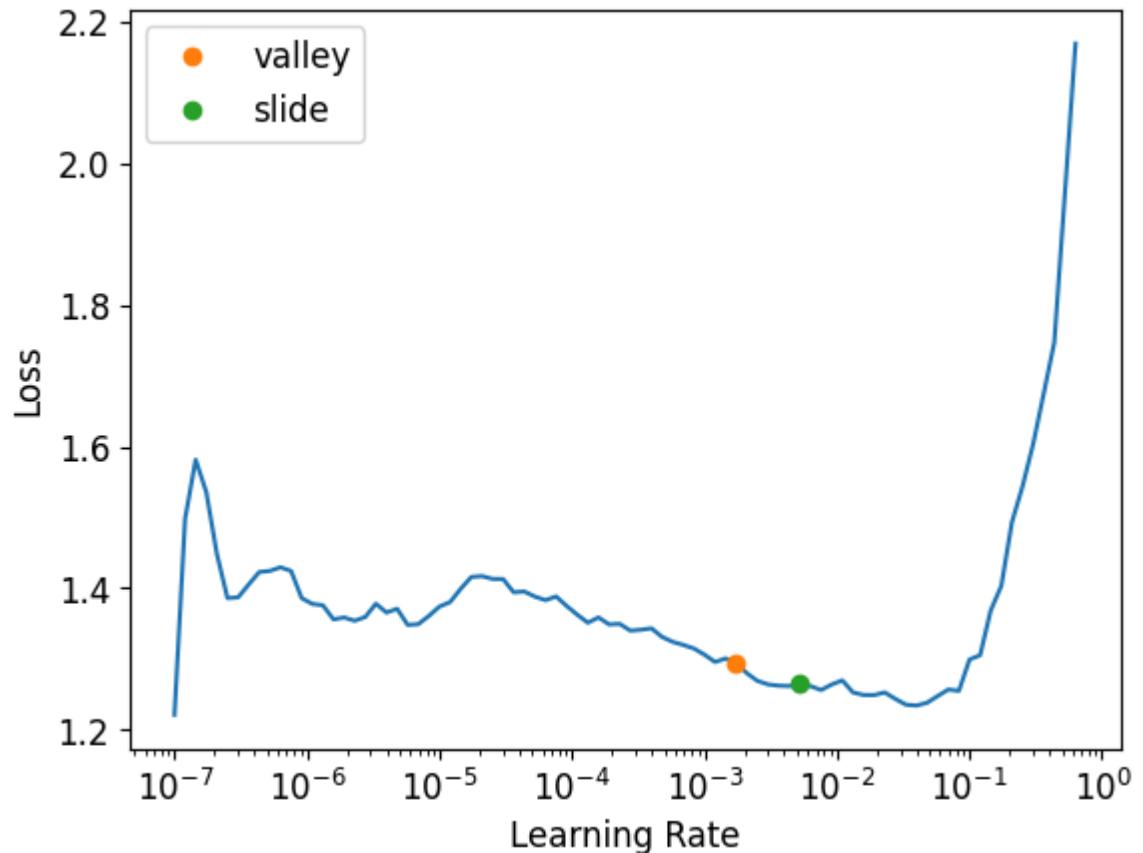
We define `learn1` series for `resnet18`.

Firstly we are going to find the suitable learning rate for the models to train.

```
In [ ]: learn1_1, dls1_1, lr1_1 = find_learning_rate_squish(resnet18)
```

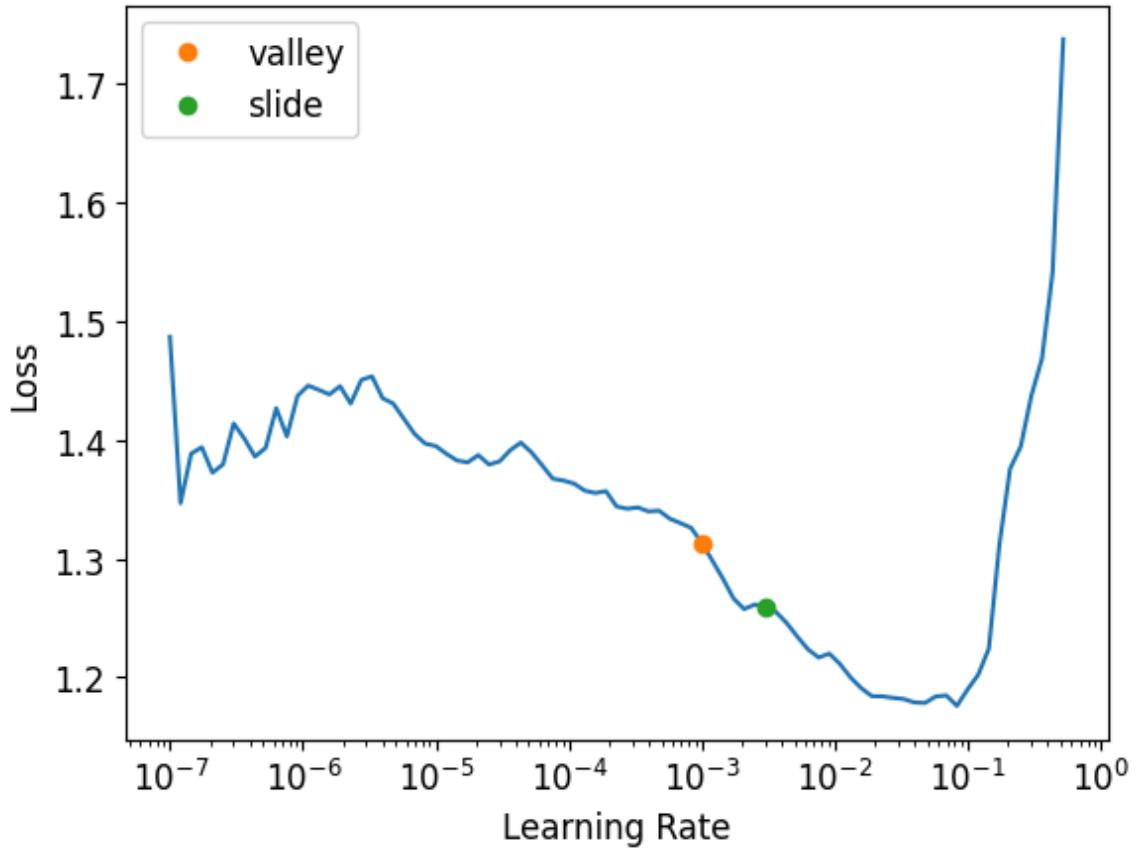
Downloading: "https://download.pytorch.org/models/resnet18-f37072fd.pth" to /root/.cache/torch/hub/checkpoints/resnet18-f37072fd.pth
100%|██████████| 44.7M/44.7M [00:00<00:00, 176MB/s]

Valley: 0.001737800776027143; Slide: 0.005248074419796467



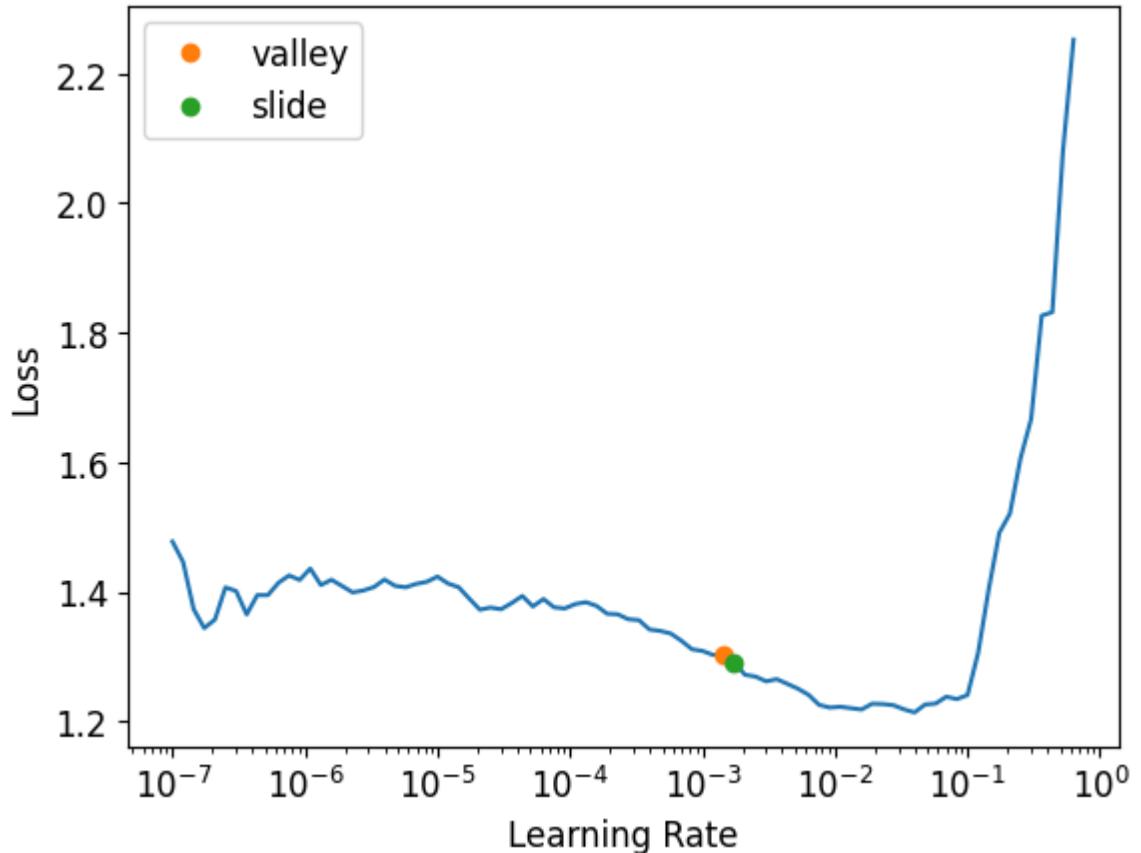
```
In [ ]: learn1_2, dls1_2, lr1_2 = find_learning_rate_zero(resnet18)
```

Valley: 0.0010000000474974513; Slide: 0.0030199517495930195



```
In [ ]: learn1_3, dls1_3, lr1_3 = datablock_method(resnet18)
```

Valley: 0.0014454397605732083; Slide: 0.001737800776027143



```
In [ ]: learn1_1.fine_tune(epochs=15, base_lr=lr1_1, freeze_epochs=3)
```

```
name1_1 = 'resnet18-squish.csv'  
desc1_1 = 'resnet18-squish-training'  
write_to_submission(dls1_1, learn1_1, ss, name1_1, desc1_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.611700	0.548581	0.281021	00:51
1	0.499814	0.475075	0.220559	00:51
2	0.423392	0.397945	0.178560	00:51

epoch	train_loss	valid_loss	error_rate	time
0	0.294185	0.299396	0.124168	00:54
1	0.257801	0.279449	0.113273	00:54
2	0.246745	0.259316	0.105705	00:54
3	0.231887	0.285710	0.118846	00:54
4	0.202739	0.315331	0.125166	00:55
5	0.187559	0.257624	0.102046	00:54
6	0.157565	0.202648	0.078759	00:54
7	0.134712	0.234977	0.081919	00:54
8	0.105739	0.250937	0.082335	00:54
9	0.092305	0.227710	0.078094	00:54
10	0.059342	0.287818	0.082834	00:54
11	0.044787	0.273216	0.074102	00:54
12	0.035923	0.274681	0.070941	00:54
13	0.034389	0.289172	0.071357	00:55
14	0.027033	0.286794	0.070027	00:54

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 139kB/s]
```

Here is the score of this model we achieved in Kaggle, to the first one, the private score is 0.76022 and the public score is 0.77840.

 resnet18-squish.csv	0.76022	0.77840	
Complete (after deadline) · 2d ago · resnet18-squish-training			

```
In [ ]: learn1_2.fine_tune(epochs=15, base_lr=lr1_2, freeze_epochs=3)
```

```
name1_2 = 'resnet18-zero-padding.csv'  
desc1_2 = 'resnet18-zero-padding-training'  
write_to_submission(dls1_2, ss, name1_2, desc1_2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.631200	0.488654	0.224967	00:50
1	0.426048	0.506552	0.238939	00:51
2	0.334894	0.504800	0.231870	00:50

epoch	train_loss	valid_loss	error_rate	time
0	0.222565	0.448485	0.201514	00:53
1	0.215481	0.456549	0.206254	00:53
2	0.198327	0.520542	0.219228	00:53
3	0.179354	0.517501	0.224301	00:53
4	0.159194	0.399289	0.170908	00:53
5	0.133714	0.435746	0.172488	00:53
6	0.139919	0.373016	0.142964	00:53
7	0.112150	0.247608	0.097139	00:53
8	0.087288	0.384073	0.135978	00:53
9	0.086766	0.374709	0.129408	00:53
10	0.070008	0.370230	0.119511	00:53
11	0.060207	0.339175	0.108283	00:53
12	0.047151	0.321024	0.101630	00:53
13	0.045579	0.346636	0.105123	00:53
14	0.038424	0.345132	0.102961	00:53

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478  
485 pixels, could be decompression bomb DOS attack.  
    warnings.warn()  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478  
485 pixels, could be decompression bomb DOS attack.  
    warnings.warn()  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478  
485 pixels, could be decompression bomb DOS attack.  
    warnings.warn()  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478  
485 pixels, could be decompression bomb DOS attack.  
    warnings.warn()  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478  
485 pixels, could be decompression bomb DOS attack.  
    warnings.warn()  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478  
485 pixels, could be decompression bomb DOS attack.  
    warnings.warn()  
100%|██████████| 52.6k/52.6k [00:00<00:00, 101kB/s]
```

This is the zero-padding method trained model, the private score is 0.90863 and the public score is 0.91321, the score for this model is quite high and I watched the leaderboard of this competition in Kaggle, this score can receive a very good score.

	resnet18-zero-padding.csv	0.90863	0.91321	<input type="checkbox"/>
	Complete (after deadline) · 2d ago · resnet18-zero-padding-training			

Here is the Leaderboard of this competition, the second highest score is 0.90307 and mine is a little higher than the second score.

#	△	Team	Members	Score	Entries	Last	Solution
1	—	Pranjal Goyal		0.95911	4	2mo	
2	—	deep_buha_6		0.90307	4	2mo	
3	—	Rahul Khichar IITGN		0.89298	16	2mo	
4	—	Aayush Kuloor		0.84048	8	2mo	
5	—	Sunay Desai		0.83341	1	2mo	
6	—	Arpan Gupta		0.79454	1	2mo	

```
In [ ]: learn1_3.fine_tune(epochs=15, base_lr=lr1_3, freeze_epochs=3)
```

```
name1_3 = 'resnet18-datablock.csv'
desc1_3 = 'resnet18-datablock-training'
write_to_submission(dls1_3, ss, name1_3, desc1_3)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.685708	0.557383	0.276281	00:42
1	0.524075	0.494951	0.244095	00:42
2	0.464741	0.425180	0.192116	00:42

epoch	train_loss	valid_loss	error_rate	time
0	0.308627	0.290878	0.122921	00:44
1	0.219584	0.252963	0.101630	00:44
2	0.163779	0.276007	0.107119	00:44
3	0.112236	0.294867	0.096723	00:44
4	0.101576	0.280545	0.090319	00:44
5	0.067960	0.306113	0.091900	00:44
6	0.043669	0.309487	0.083666	00:44
7	0.035009	0.319854	0.080755	00:44
8	0.026601	0.342042	0.080506	00:44
9	0.013237	0.388655	0.080506	00:44
10	0.011681	0.386378	0.079591	00:44
11	0.004147	0.397884	0.076098	00:44
12	0.003178	0.397511	0.073603	00:44
13	0.002547	0.400401	0.073021	00:44
14	0.003691	0.403836	0.073436	00:44

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 101kB/s]
```

This is the score for the third model. We find that the score is much lower than the zero-padding method but it is a little higher than the first one.

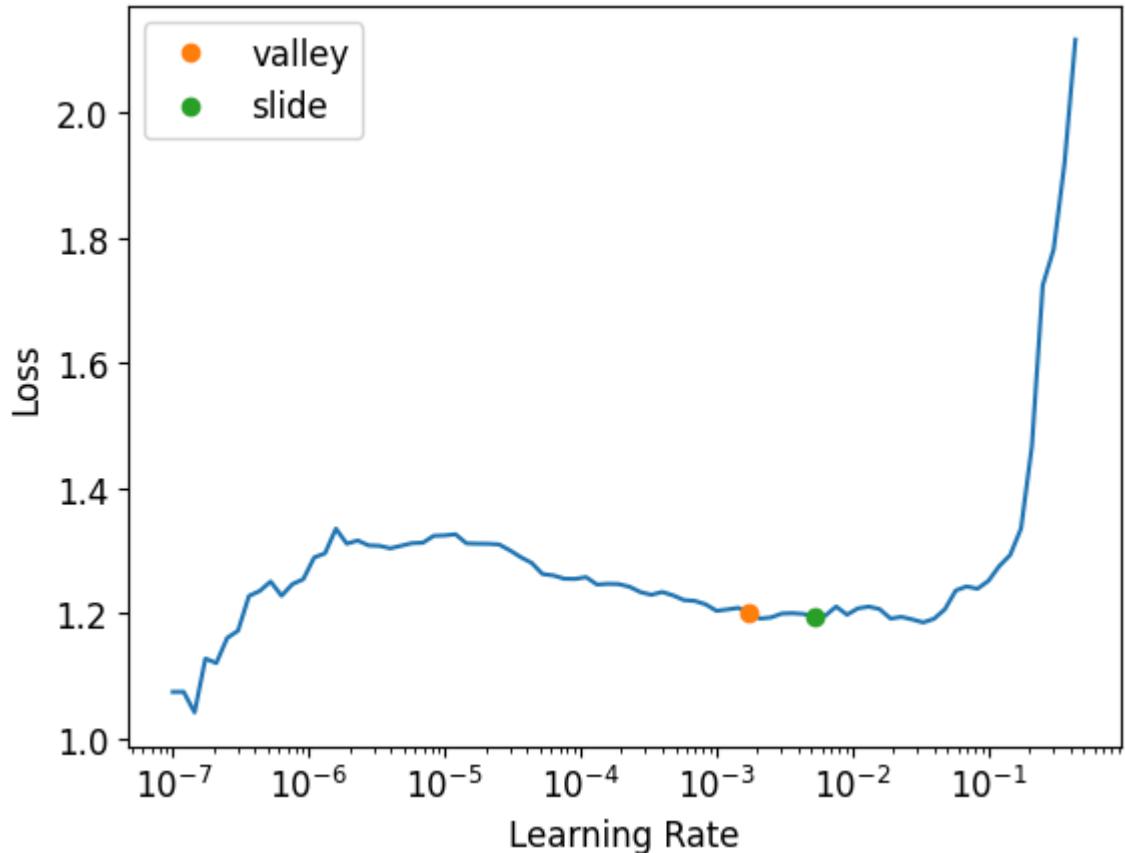
 resnet18-datablock.csv	0.78142	0.79496	
Complete (after deadline) · 2d ago · resnet18-datablock-training			

We use `learn2` series for `resnet34`. Also, we need to find a suitable learning rate for the model training.

```
In [ ]: learn2_1, dls2_1, lr2_1 = find_learning_rate_squish(resnet34)
```

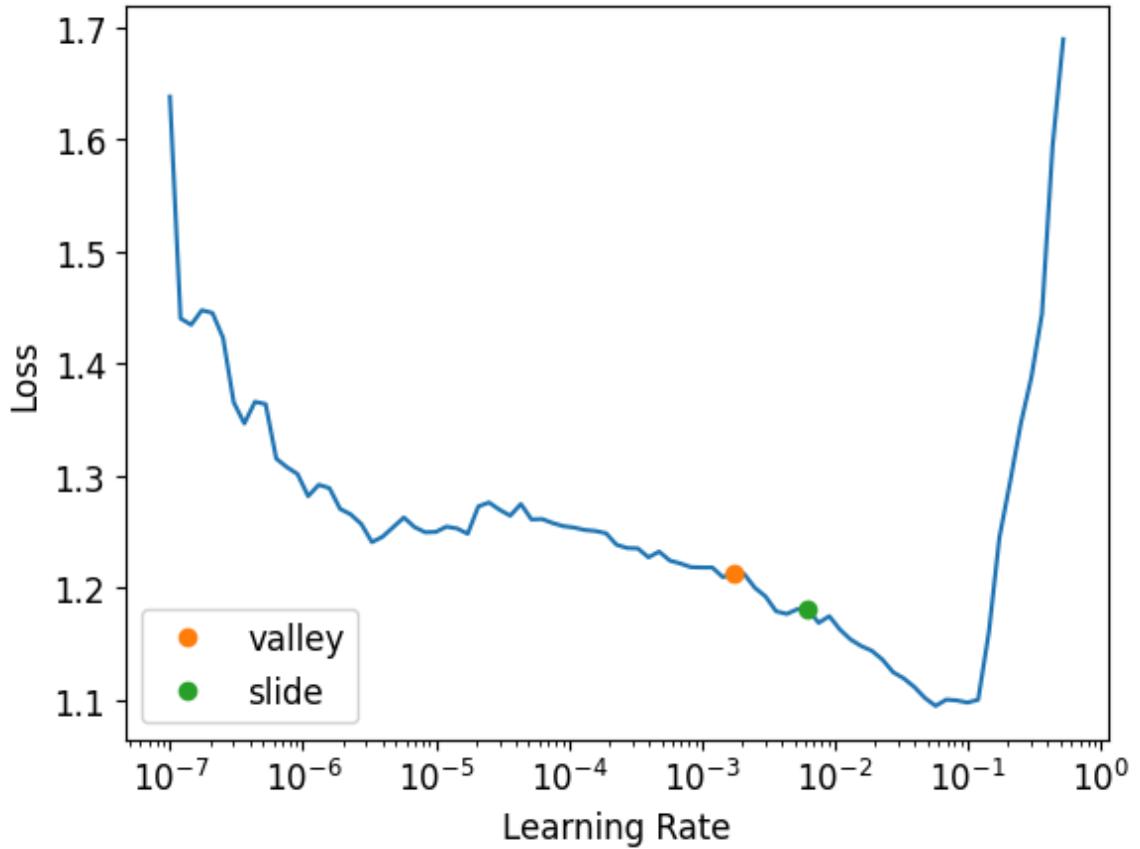
Downloading: "https://download.pytorch.org/models/resnet34-b627a593.pth" to /root/.cache/torch/hub/checkpoints/resnet34-b627a593.pth
100%|██████████| 83.3M/83.3M [00:00<00:00, 180MB/s]

Valley: 0.001737800776027143; Slide: 0.005248074419796467



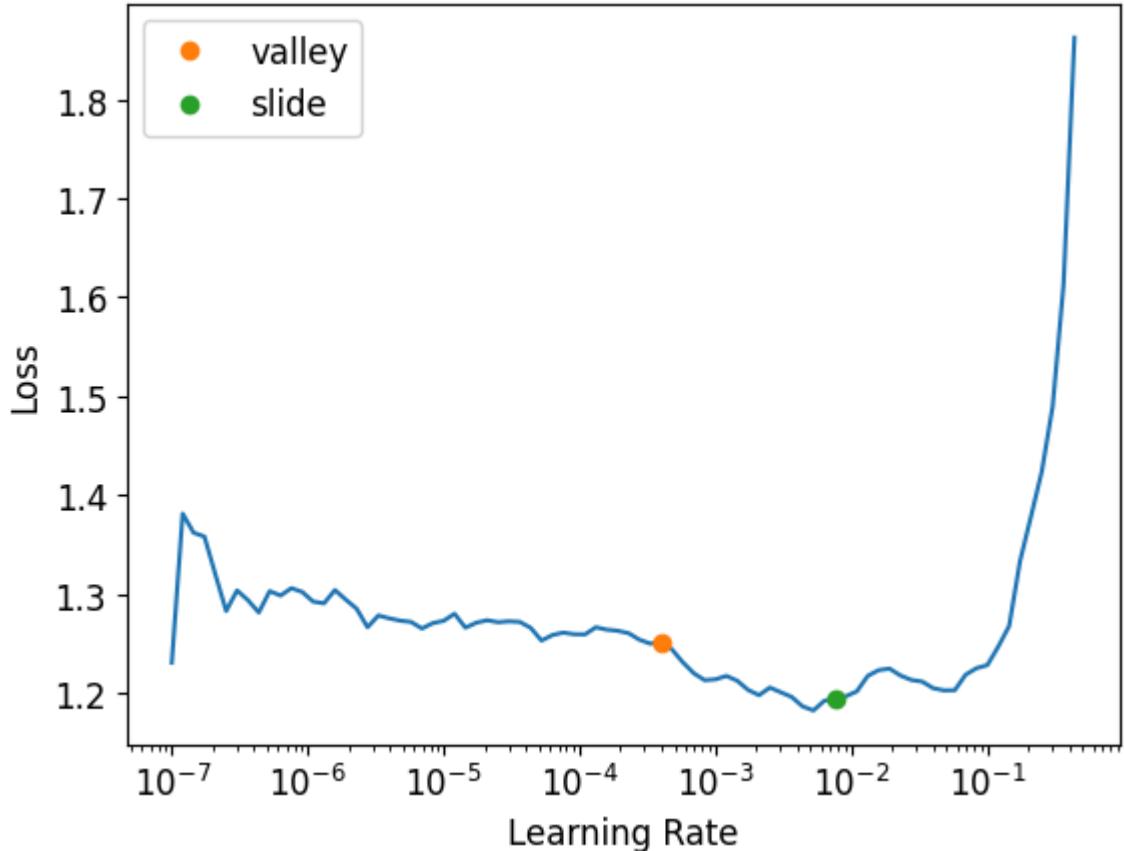
```
In [ ]: learn2_2, dls2_2, lr2_2 = find_learning_rate_zero(resnet34)
```

Valley: 0.001737800776027143; Slide: 0.0063095735386013985



```
In [ ]: learn2_3, dls2_3, lr2_3 = datablock_method(resnet34)
```

Valley: 0.0003981071640737355; Slide: 0.007585775572806597



```
In [ ]: learn2_1.fine_tune(epochs=15, base_lr=lr2_1, freeze_epochs=3)
```

```
name2_1 = 'resnet34-squish.csv'
desc2_1 = 'resnet34-squish-training'
write_to_submission(dls2_1, learn2_1, ss, name2_1, desc2_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.601061	0.522776	0.265552	00:58
1	0.457408	0.443767	0.197023	00:59
2	0.383127	0.353869	0.149701	00:58

epoch	train_loss	valid_loss	error_rate	time
0	0.273475	0.264378	0.107036	01:03
1	0.235790	0.322401	0.123087	01:03
2	0.232766	0.221049	0.084082	01:03
3	0.209448	0.231148	0.090985	01:03
4	0.188834	0.198824	0.076347	01:04
5	0.164742	0.239739	0.089820	01:04
6	0.143037	0.187161	0.068114	01:04
7	0.116519	0.253724	0.092232	01:04
8	0.086796	0.228887	0.071108	01:03
9	0.073734	0.209805	0.067448	01:04
10	0.045843	0.227586	0.062708	01:04
11	0.033978	0.247061	0.063290	01:04
12	0.023326	0.245718	0.061211	01:03
13	0.021256	0.269248	0.060546	01:04
14	0.014376	0.256603	0.058550	01:03

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 149kB/s]
```

This is the score for resnet34 of squish method.

 resnet34-squish.csv	0.76274	0.78039	<input type="checkbox"/>
Complete (after deadline) · 2d ago · resnet34-squish-training			

```
In [ ]: learn2_2.fine_tune(epochs=15, base_lr=lr2_2, freeze_epochs=3)
```

```
name2_2 = 'resnet34-zero-paddding.csv'  
desc2_2 = 'resnet34-zero-paddding-training'  
write_to_submission(dls2_2, ss, name2_2, desc2_2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.510880	0.465360	0.216733	00:58
1	0.354900	0.555963	0.254907	00:58
2	0.285147	0.505455	0.226297	00:57

epoch	train_loss	valid_loss	error_rate	time
0	0.209456	0.478564	0.219311	01:03
1	0.191582	0.320368	0.141800	01:02
2	0.189541	0.461159	0.179391	01:02
3	0.172515	0.359833	0.159930	01:02
4	0.164915	0.519773	0.240269	01:02
5	0.145852	0.411561	0.171075	01:02
6	0.118814	0.381954	0.144711	01:02
7	0.114574	0.307260	0.120842	01:02
8	0.097049	0.353458	0.135479	01:02
9	0.087620	0.333815	0.122255	01:02
10	0.072581	0.400654	0.145625	01:03
11	0.058483	0.379518	0.124834	01:02
12	0.044312	0.368910	0.116018	01:03
13	0.037559	0.373874	0.107868	01:02
14	0.034103	0.353888	0.103127	01:02

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 102kB/s]
```

This is the resnet34 with zero-padding method, we can find that the score for this model is also high. Though it has a little decrease in accuracy compare to the same method trained with resnet18.

 resnet34-zero-paddding.csv	0.90156	0.90261	
Complete (after deadline) · 2d ago · resnet34-zero-paddding-training			

```
In [ ]: learn2_3.fine_tune(epochs=15, base_lr=lr2_3, freeze_epochs=3)
```

```
name2_3 = 'resnet34-datablock.csv'  
desc2_3 = 'resnet34-datablock-training'  
write_to_submission(dls2_3, learn2_3, ss, name2_3, desc2_3)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.563362	0.511644	0.254075	00:49
1	0.430513	0.404144	0.176896	00:48
2	0.347845	0.301892	0.129907	00:49

epoch	train_loss	valid_loss	error_rate	time
0	0.225978	0.216199	0.089321	00:53
1	0.194038	0.200919	0.082834	00:54
2	0.180232	0.216197	0.087159	00:54
3	0.151792	0.283667	0.105206	00:53
4	0.131358	0.197803	0.077179	00:54
5	0.086846	0.256312	0.085995	00:53
6	0.062165	0.255094	0.078759	00:54
7	0.038761	0.254742	0.070110	00:53
8	0.020237	0.283911	0.071025	00:53
9	0.016170	0.263906	0.059464	00:54
10	0.008788	0.328505	0.064039	00:54
11	0.006450	0.319917	0.058633	00:54
12	0.001060	0.306868	0.053393	00:54
13	0.000820	0.322114	0.055722	00:54
14	0.000312	0.320233	0.055306	00:53

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 96.1kB/s]
```

The resnet34 trained with datablock method.

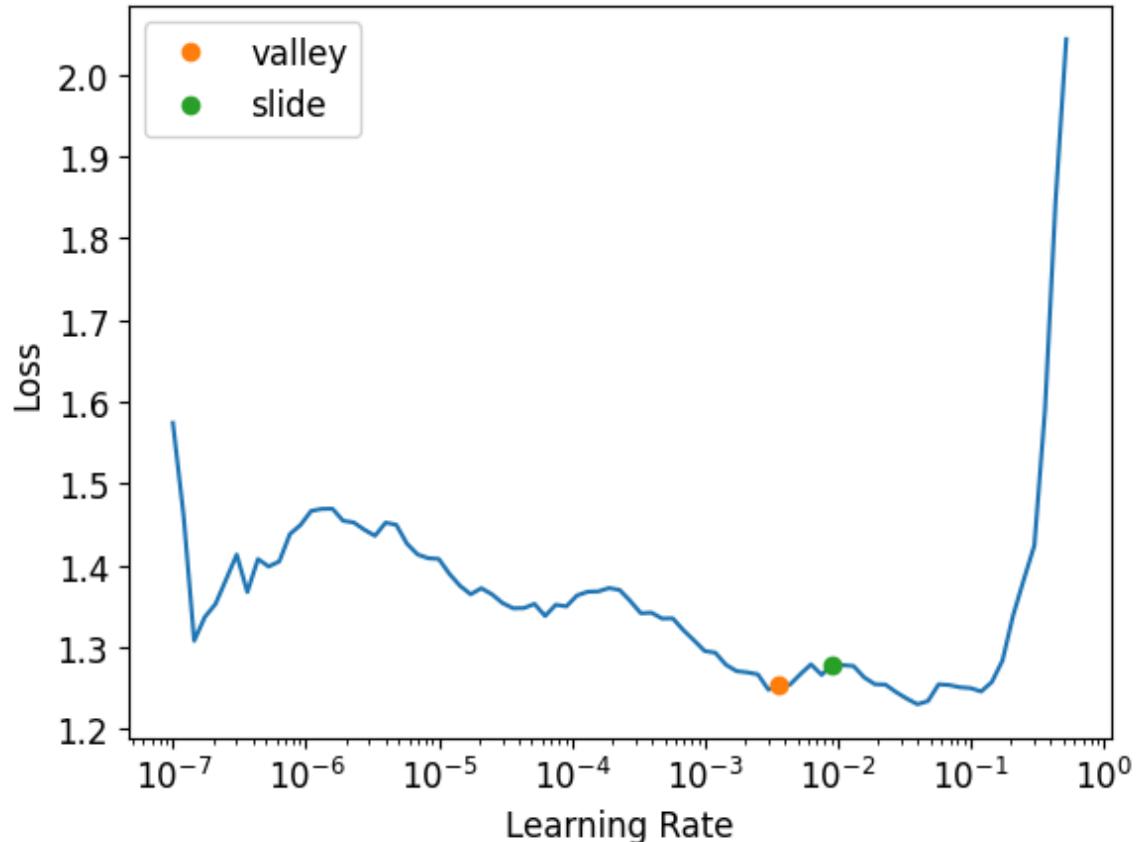
 resnet34-datablock.csv	0.76981	0.78072	
Complete (after deadline) · 2d ago · resnet34-datablock-training			

We define learn3 series for resnet50 .

```
In [ ]: learn3_1, dls3_1, lr3_1 = find_learning_rate_squish(resnet50)
```

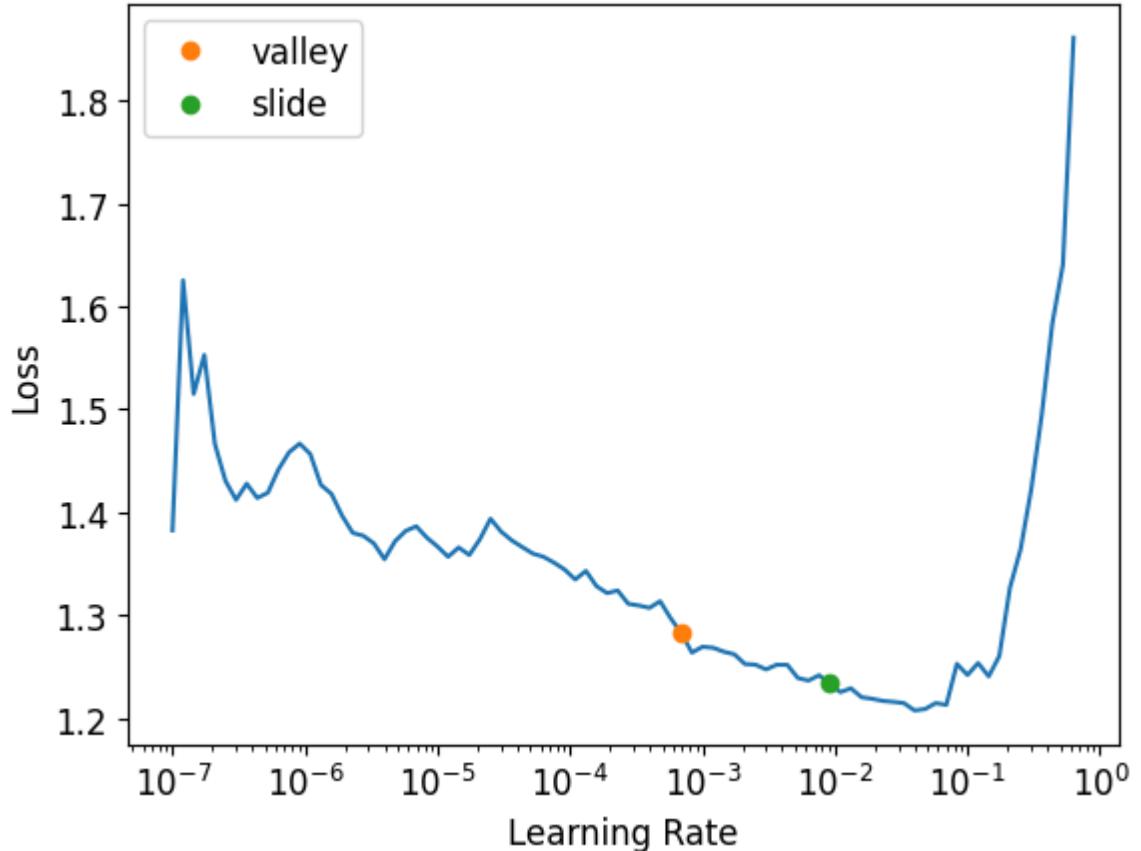
Downloading: "https://download.pytorch.org/models/resnet50-11ad3fa6.pth" to /root/.cache/torch/hub/checkpoints/resnet50-11ad3fa6.pth
100%|██████████| 97.8M/97.8M [00:00<00:00, 195MB/s]

Valley: 0.00363078061491251; Slide: 0.009120108559727669



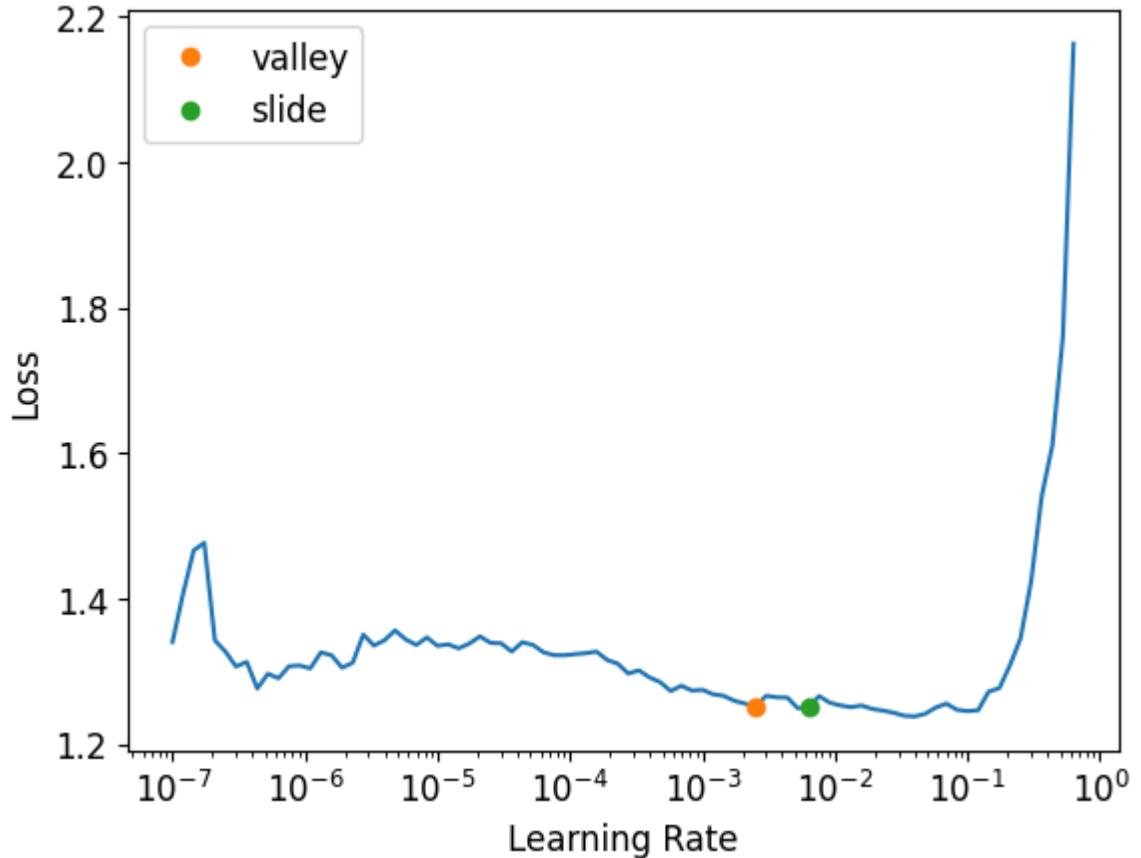
```
In [ ]: learn3_2, dls3_2, lr3_2 = find_learning_rate_zero(resnet50)
```

Valley: 0.0006918309954926372; Slide: 0.009120108559727669



```
In [ ]: learn3_3, dls3_3, lr3_3 = datablock_method(resnet50)
```

Valley: 0.002511886414140463; Slide: 0.0063095735386013985



```
In [ ]: learn3_1.fine_tune(epochs=15, base_lr=lr3_1, freeze_epochs=3)
```

```
name3_1 = 'resnet50-squish.csv'
desc3_1 = 'resnet50-squish-training'
write_to_submission(dls3_1, learn3_1, ss, name3_1, desc3_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.484963	0.462493	0.214904	01:08
1	0.408274	0.358479	0.152861	01:08
2	0.313036	0.295044	0.126747	01:08

epoch	train_loss	valid_loss	error_rate	time
0	0.194764	0.181260	0.072688	01:19
1	0.153257	0.206140	0.078426	01:19
2	0.162386	0.237074	0.094145	01:19
3	0.148957	0.200095	0.079591	01:19
4	0.138500	0.166009	0.064621	01:19
5	0.109709	0.170239	0.062292	01:20
6	0.107382	0.182876	0.071108	01:19
7	0.089446	0.202830	0.076680	01:19
8	0.058261	0.166404	0.052229	01:19
9	0.053837	0.190041	0.060296	01:19
10	0.041615	0.164313	0.048736	01:19
11	0.021457	0.191120	0.049318	01:20
12	0.014865	0.195878	0.047322	01:19
13	0.014719	0.183641	0.044494	01:19
14	0.012725	0.183407	0.045742	01:19

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 110kB/s]
```

The score for `resnet50` in squish method.

 resnet50-squish.csv	0.72993	0.73534	<input type="checkbox"/>
Complete (after deadline) · 2d ago · resnet50-squish-training			

```
In [ ]: learn3_2.fine_tune(epochs=15, base_lr=lr3_2, freeze_epochs=3)
```

```
name3_2 = 'resnet50-zero-paddding.csv'  
desc3_2 = 'resnet50-zero-paddding-training'  
write_to_submission(dls3_2, ss, name3_2, desc3_2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.456529	0.496563	0.236527	01:04
1	0.386574	0.492383	0.231121	01:05
2	0.295751	0.322151	0.133899	01:05

epoch	train_loss	valid_loss	error_rate	time
0	0.180351	0.214022	0.089238	01:12
1	0.141814	0.182760	0.071524	01:11
2	0.132050	0.160266	0.059132	01:12
3	0.117113	0.157314	0.059880	01:11
4	0.110930	0.160033	0.058716	01:12
5	0.099868	0.148192	0.055057	01:11
6	0.091667	0.189412	0.069278	01:12
7	0.079736	0.186816	0.065536	01:12
8	0.066426	0.169678	0.059963	01:12
9	0.040438	0.153767	0.045409	01:12
10	0.035128	0.186212	0.047488	01:12
11	0.025764	0.223339	0.058051	01:12
12	0.017691	0.189297	0.046075	01:12
13	0.015506	0.172960	0.039338	01:12
14	0.019995	0.192767	0.042166	01:12

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 145kB/s]
```



resnet50-zero-paddning.csv

Complete (after deadline) · 2d ago · resnet50-zero-paddning-training

0.86572

0.87280



```
In [ ]: learn3_3.fine_tune(epochs=15, base_lr=lr3_3, freeze_epochs=3)
```

```
name3_3 = 'resnet50-datablock.csv'  
desc3_3 = 'resnet50-datablock-training'  
write_to_submission(dls3_3, learn3_3, ss, name3_3, desc3_3)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.480670	0.423808	0.195110	00:57
1	0.394369	0.336195	0.146956	00:57
2	0.312199	0.265659	0.106121	00:57

epoch	train_loss	valid_loss	error_rate	time
0	0.161963	0.147233	0.055057	01:07
1	0.100725	0.137674	0.048819	01:07
2	0.089194	0.139949	0.050649	01:07
3	0.091468	0.141188	0.050067	01:07
4	0.078059	0.132072	0.044494	01:07
5	0.059020	0.142074	0.048237	01:07
6	0.045452	0.172436	0.045326	01:07
7	0.030042	0.147354	0.040669	01:07
8	0.024437	0.142189	0.037591	01:07
9	0.019912	0.140843	0.036593	01:07
10	0.005914	0.178060	0.033766	01:06
11	0.006046	0.170704	0.034764	01:07
12	0.004060	0.173129	0.033516	01:07
13	0.001120	0.161683	0.031021	01:07
14	0.001650	0.167636	0.031603	01:07

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 96.7kB/s]
```

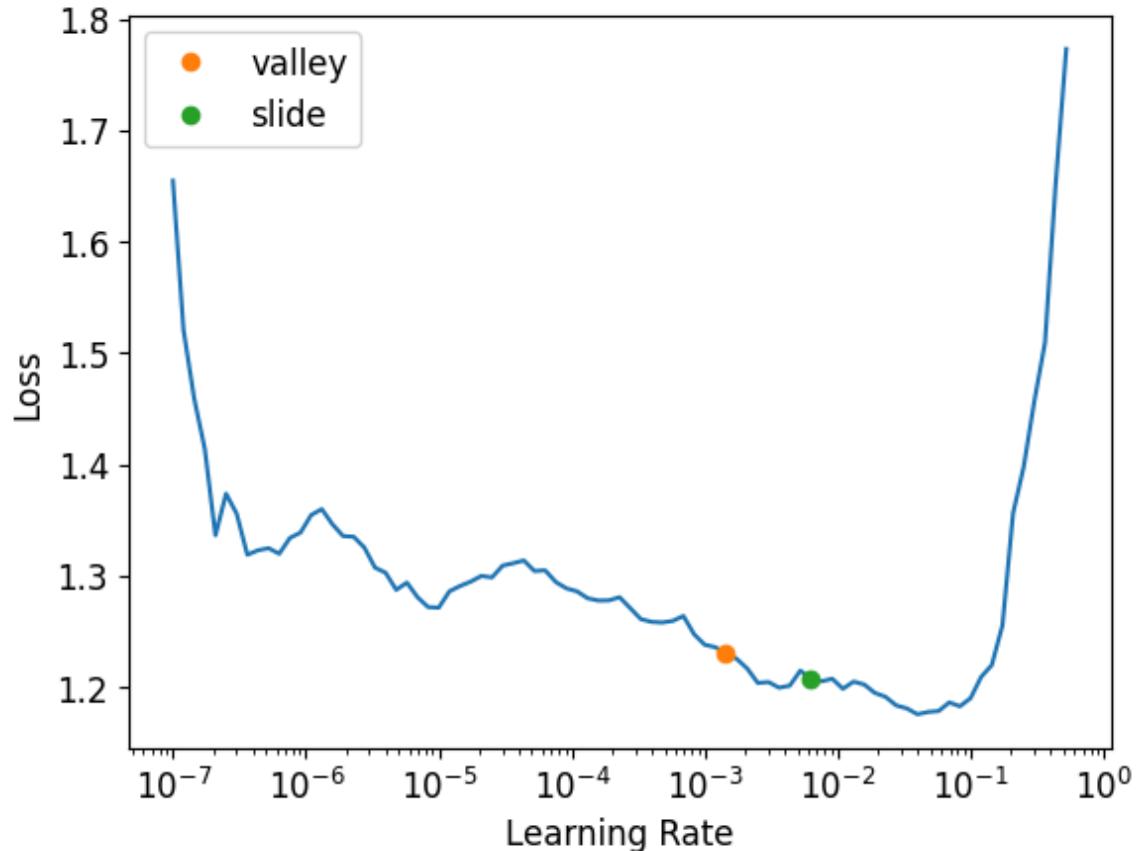
 resnet50-datablock.csv	0.69207	0.69493	
Complete (after deadline) · 2d ago · resnet50-datablock-training			

We define `learn4` series for `resnet101`.

```
In [ ]: learn4_1, dls4_1, lr4_1 = find_learning_rate_squish(resnet101)
```

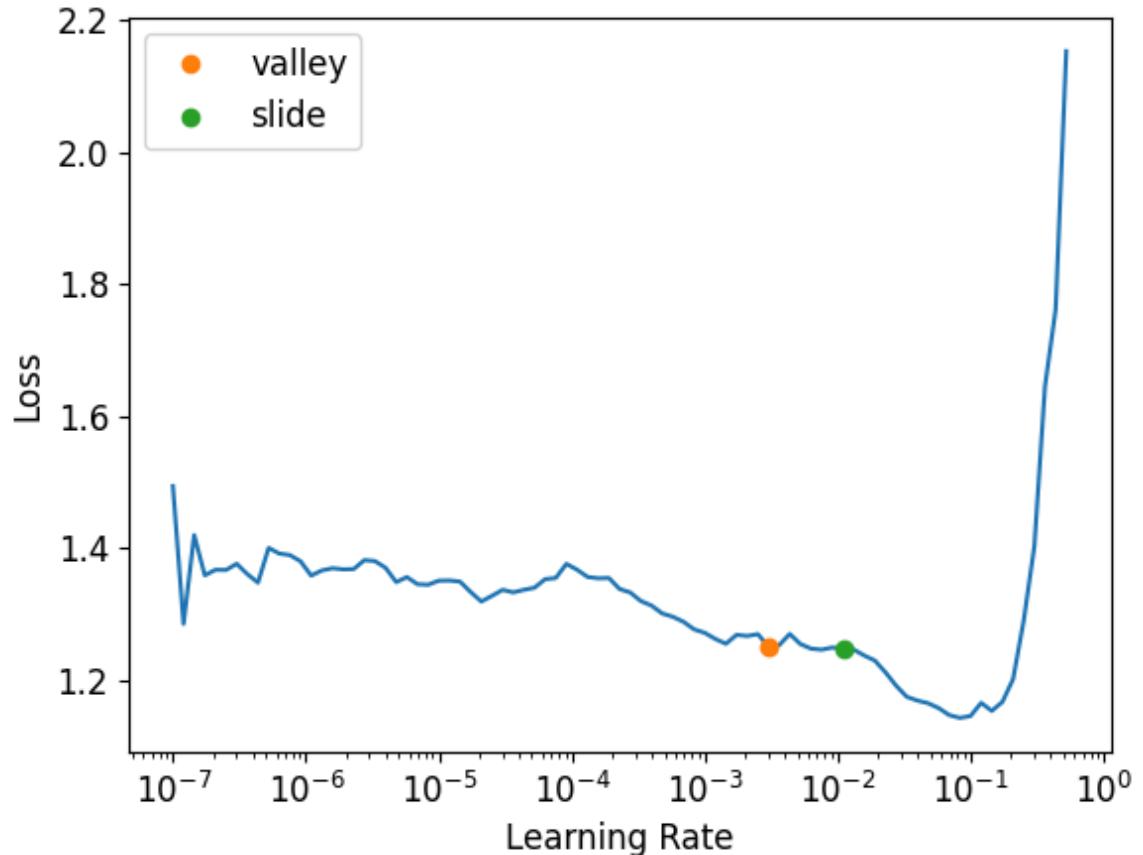
Downloading: "https://download.pytorch.org/models/resnet101-cd907fc2.pth" to /root/.cache/torch/hub/checkpoints/resnet101-cd907fc2.pth
100%|██████████| 171M/171M [00:00<00:00, 208MB/s]

Valley: 0.0014454397605732083; Slide: 0.0063095735386013985



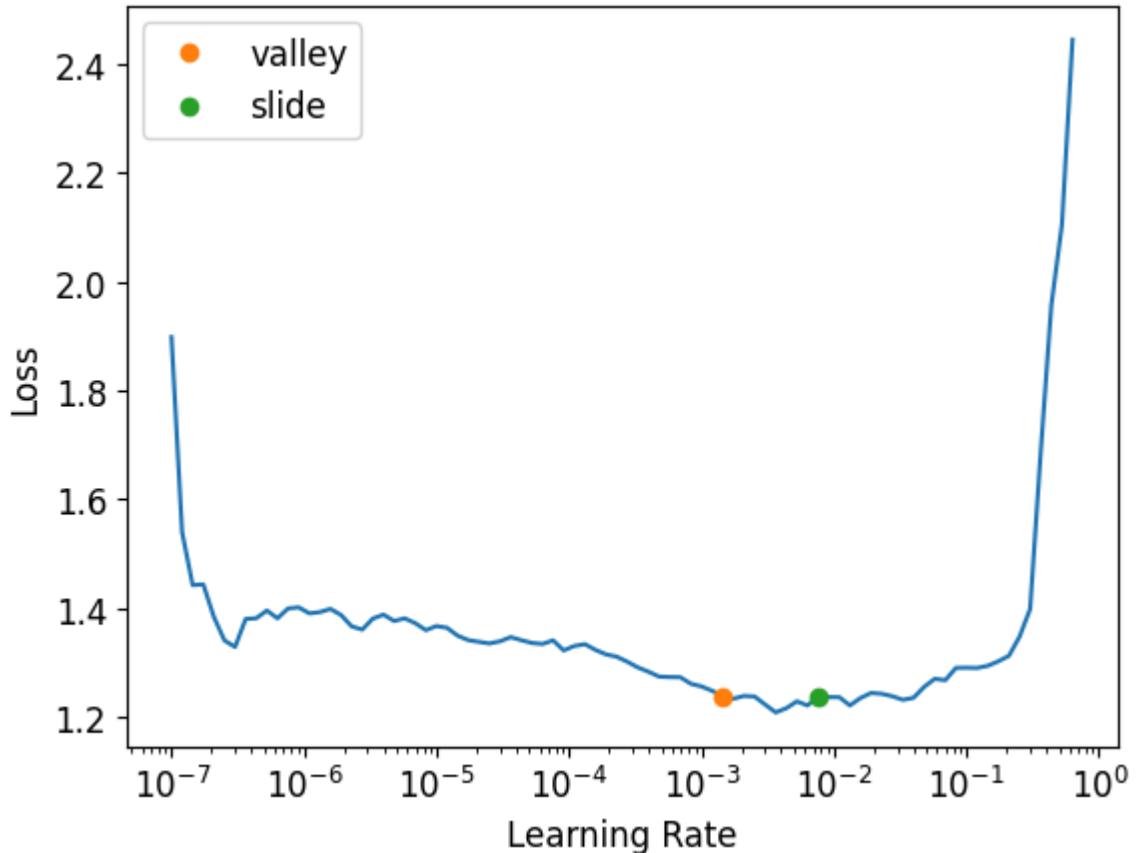
```
In [ ]: learn4_2, dls4_2, lr4_2 = find_learning_rate_zero(resnet101)
```

Valley: 0.0030199517495930195; Slide: 0.010964781977236271



```
In [ ]: learn4_3, dls4_3, lr4_3 = datablock_method(resnet101)
```

Valley: 0.0014454397605732083; Slide: 0.007585775572806597



```
In [ ]: learn4_1.fine_tune(epochs=15, base_lr=lr4_1, freeze_epochs=3)
```

```
name4_1 = 'resnet101-squish.csv'
desc4_1 = 'resnet101-squish-training'
write_to_submission(dls4_1, learn4_1, ss, name4_1, desc4_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.531368	0.483656	0.230872	01:32
1	0.418946	0.377711	0.170077	01:32
2	0.346721	0.311509	0.132152	01:32

epoch	train_loss	valid_loss	error_rate	time
0	0.182372	0.178498	0.070858	01:57
1	0.149992	0.183361	0.069278	01:58
2	0.144148	0.181086	0.067947	01:58
3	0.131402	0.243951	0.091983	01:58
4	0.121940	0.213681	0.080506	01:58
5	0.100570	0.205246	0.075682	01:58
6	0.081407	0.189325	0.065702	01:58
7	0.068966	0.186230	0.063041	01:58
8	0.038513	0.206464	0.058882	01:58
9	0.035441	0.190924	0.053809	01:58
10	0.021691	0.230937	0.053227	01:58
11	0.016830	0.196532	0.046989	01:58
12	0.009349	0.194821	0.043912	01:58
13	0.009019	0.202610	0.045326	01:58
14	0.006119	0.193765	0.043496	01:58

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 141kB/s]
```



resnet101-squish.csv

Complete (after deadline) · 2d ago · resnet101-squish-training

0.73346

0.74826



```
In [ ]: learn4_2.fine_tune(epochs=15, base_lr=lr4_2, freeze_epochs=3)
```

```
name4_2 = 'resnet101-zero-padding.csv'  
desc4_2 = 'resnet101-zero-padding-training'  
write_to_submission(dls4_2, ss, name4_2, desc4_2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.459211	0.515613	0.255073	01:28
1	0.360699	0.541563	0.232119	01:28
2	0.276863	0.334114	0.141218	01:29

epoch	train_loss	valid_loss	error_rate	time
0	0.162636	0.236127	0.096391	01:44
1	0.138451	0.305376	0.122754	01:43
2	0.157482	0.356354	0.140635	01:44
3	0.144957	0.234416	0.103044	01:44
4	0.138993	0.290560	0.117349	01:44
5	0.106361	0.352458	0.132818	01:44
6	0.101351	0.327606	0.125333	01:45
7	0.085641	0.323549	0.115436	01:44
8	0.064952	0.306470	0.109115	01:44
9	0.048563	0.175345	0.056886	01:45
10	0.037529	0.260228	0.059049	01:45
11	0.030735	0.270452	0.080173	01:44
12	0.023872	0.206662	0.060130	01:44
13	0.017980	0.181364	0.049734	01:45
14	0.020438	0.214344	0.057468	01:44

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 101kB/s]
```



resnet101-zero-paddng.csv

Complete (after deadline) · 2d ago · resnet101-zero-paddng-training

0.86017

0.87379



```
In [ ]: learn4_3.fine_tune(epochs=15, base_lr=lr4_3, freeze_epochs=3)
```

```
name4_3 = 'resnet101-datablock.csv'  
desc4_3 = 'resnet101-datablock-training'  
write_to_submission(dls4_3, ss, name4_3, desc4_3)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.497242	0.436711	0.204258	01:20
1	0.399787	0.322667	0.140220	01:19
2	0.293478	0.255148	0.103044	01:20

epoch	train_loss	valid_loss	error_rate	time
0	0.158169	0.141723	0.055556	01:45
1	0.102165	0.160117	0.057552	01:46
2	0.102864	0.148624	0.052229	01:45
3	0.103062	0.180535	0.062957	01:46
4	0.094237	0.144701	0.051480	01:45
5	0.071970	0.132497	0.044079	01:46
6	0.045410	0.153621	0.047073	01:45
7	0.037993	0.141355	0.040419	01:45
8	0.022173	0.154773	0.039088	01:45
9	0.016559	0.153949	0.037591	01:45
10	0.004398	0.168067	0.037259	01:48
11	0.003310	0.190782	0.035013	01:48
12	0.001014	0.167292	0.031104	01:46
13	0.000808	0.170298	0.031770	01:45
14	0.000700	0.172458	0.031520	01:46

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 99.4kB/s]
```

 resnet101-datablock.csv	0.71378	0.71911	
Complete (after deadline) · 2d ago · resnet101-datablock-training			

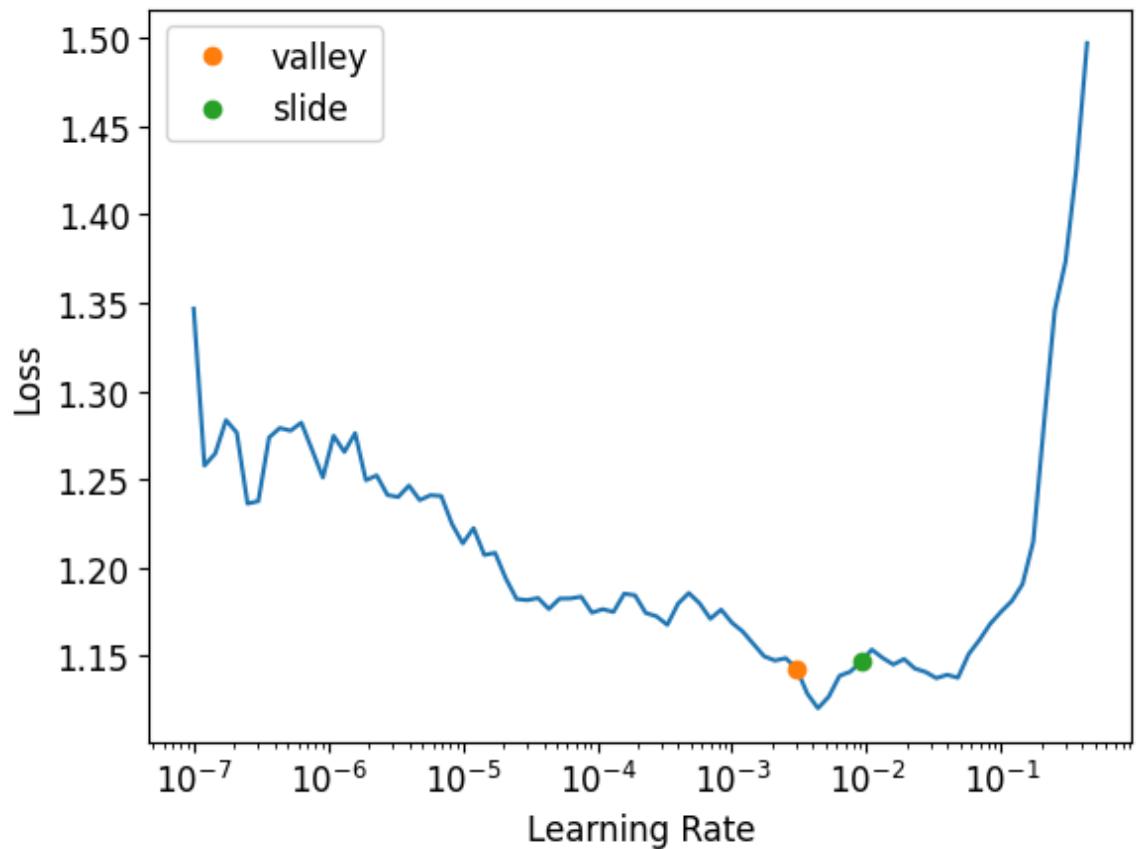
We define `learn5` series for `resnet26d`.

```
In [ ]: learn5_1, dls5_1, lr5_1 = find_learning_rate_squish(arch1)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:  
  The secret `HF_TOKEN` does not exist in your Colab secrets.  
  To authenticate with the Hugging Face Hub, create a token in your se  
  ttings tab (https://huggingface.co/settings/tokens), set it as secre  
  t in your Google Colab and restart your session.  
  You will be able to reuse this secret in all of your notebooks.  
  Please note that authentication is recommended but still optional to  
  access public models or datasets.  
    warnings.warn(
```

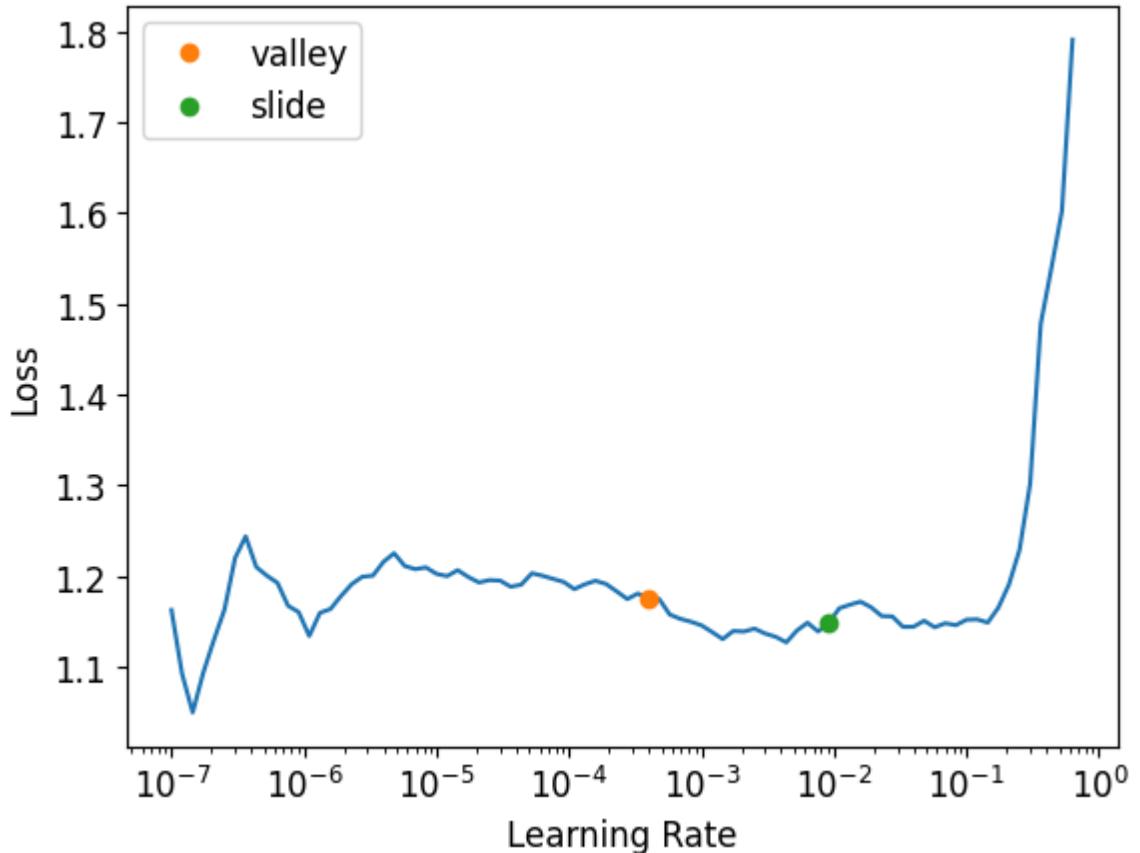
```
model.safetensors:  0%|          | 0.00/64.2M [00:00<?, ?B/s]
```

Valley: 0.0030199517495930195; Slide: 0.009120108559727669



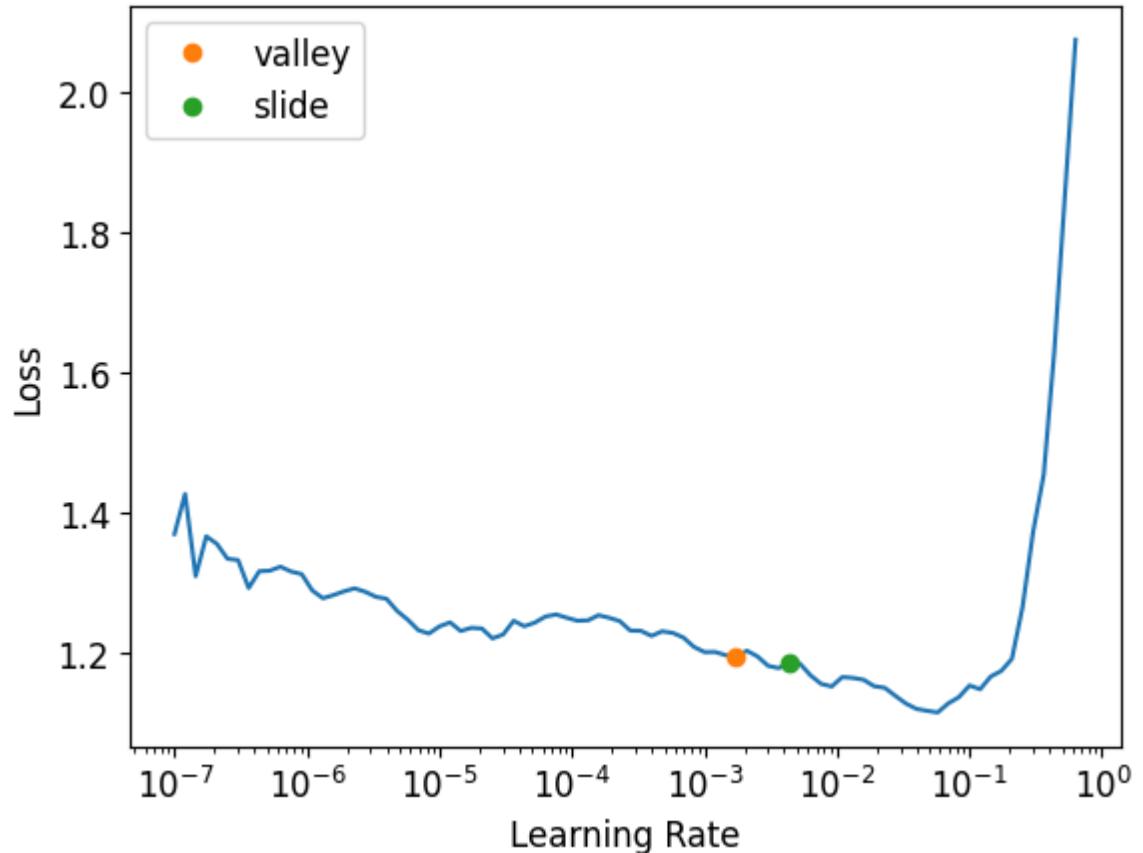
```
In [ ]: learn5_2, dls5_2, lr5_2 = find_learning_rate_zero(arch1)
```

Valley: 0.0003981071640737355; Slide: 0.009120108559727669



```
In [ ]: learn5_3, dls5_3, lr5_3 = datablock_method(arch1)
```

Valley: 0.001737800776027143; Slide: 0.004365158267319202



```
In [ ]: learn5_1.fine_tune(epochs=15, base_lr=lr5_1, freeze_epochs=3)
```

```
name5_1 = 'resnet26d-squish.csv'
desc5_1 = 'resnet26d-squish-training'
write_to_submission(dls5_1, learn5_1, ss, name5_1, desc5_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.465404	0.462163	0.217648	01:02
1	0.406606	0.398676	0.169661	01:02
2	0.344382	0.318375	0.136061	01:01

epoch	train_loss	valid_loss	error_rate	time
0	0.257574	0.250712	0.101464	01:08
1	0.225012	0.212957	0.082086	01:08
2	0.187736	0.202652	0.078510	01:08
3	0.167521	0.194165	0.071607	01:07
4	0.129438	0.186318	0.065286	01:08
5	0.110927	0.192241	0.065286	01:07
6	0.083607	0.191286	0.061710	01:07
7	0.089411	0.195603	0.066035	01:08
8	0.061127	0.217412	0.062874	01:07
9	0.049145	0.217342	0.061377	01:08
10	0.047029	0.221657	0.060712	01:08
11	0.040857	0.219452	0.059215	01:07
12	0.031095	0.217726	0.055472	01:08
13	0.026211	0.237131	0.059631	01:07
14	0.026857	0.218237	0.056387	01:08

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 103kB/s]
```



resnet26d-squish.csv

Complete (after deadline) · 2d ago · resnet26d-squish-training

0.73902

0.74229



```
In [ ]: learn5_2.fine_tune(epochs=15, base_lr=lr5_2, freeze_epochs=3)
```

```
name5_2 = 'resnet26d-zero-padding.csv'  
desc5_2 = 'resnet26d-zero-padding-training'  
write_to_submission(dls5_2, ss, name5_2, desc5_2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.423126	0.414563	0.190702	00:57
1	0.365435	0.424430	0.174152	00:57
2	0.281435	0.393245	0.155938	00:56

epoch	train_loss	valid_loss	error_rate	time
0	0.232161	0.358251	0.151198	01:01
1	0.193512	0.273721	0.115935	01:01
2	0.184428	0.275179	0.115602	01:01
3	0.161218	0.308586	0.129408	01:01
4	0.135767	0.241968	0.101880	01:00
5	0.118562	0.276878	0.106786	01:00
6	0.102124	0.263727	0.096307	01:01
7	0.094352	0.265132	0.094228	01:01
8	0.080201	0.233639	0.079923	01:01
9	0.066517	0.237166	0.079341	01:00
10	0.067834	0.222877	0.078094	01:00
11	0.052425	0.216981	0.071607	01:00
12	0.053010	0.235682	0.076514	01:00
13	0.056891	0.229174	0.072771	01:06
14	0.047685	0.259372	0.082585	01:01

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 104kB/s]
```



resnet26d-zero-paddning.csv

Complete (after deadline) · 2d ago · resnet26d-zero-paddning-training

0.88591

0.88870



```
In [ ]: learn5_3.fine_tune(epochs=15, base_lr=lr5_3, freeze_epochs=3)
```

```
name5_3 = 'resnet26d-datablock.csv'
desc5_3 = 'resnet26d-datablock-training'
write_to_submission(dls5_3, ss, name5_3, desc5_3)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.474079	0.400435	0.183051	00:48
1	0.394179	0.347437	0.153194	00:48
2	0.325651	0.282892	0.119012	00:49

epoch	train_loss	valid_loss	error_rate	time
0	0.235231	0.230282	0.094311	00:55
1	0.197065	0.206743	0.084331	00:55
2	0.157988	0.195155	0.074684	00:54
3	0.135613	0.198962	0.070193	00:54
4	0.083156	0.210503	0.067282	00:55
5	0.050257	0.224244	0.061460	00:54
6	0.042314	0.251443	0.065536	00:55
7	0.033926	0.254011	0.059880	00:55
8	0.021005	0.261827	0.059132	00:55
9	0.018685	0.272713	0.059215	00:54
10	0.016089	0.275386	0.058383	00:54
11	0.011632	0.275249	0.057136	00:55
12	0.009419	0.289869	0.058051	00:55
13	0.007857	0.292444	0.058799	00:55
14	0.007878	0.287516	0.056554	00:54

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 104kB/s]
```



resnet26d-datablock.csv

Complete (after deadline) · 2d ago · resnet26d-datablock-training

0.69914

0.71083



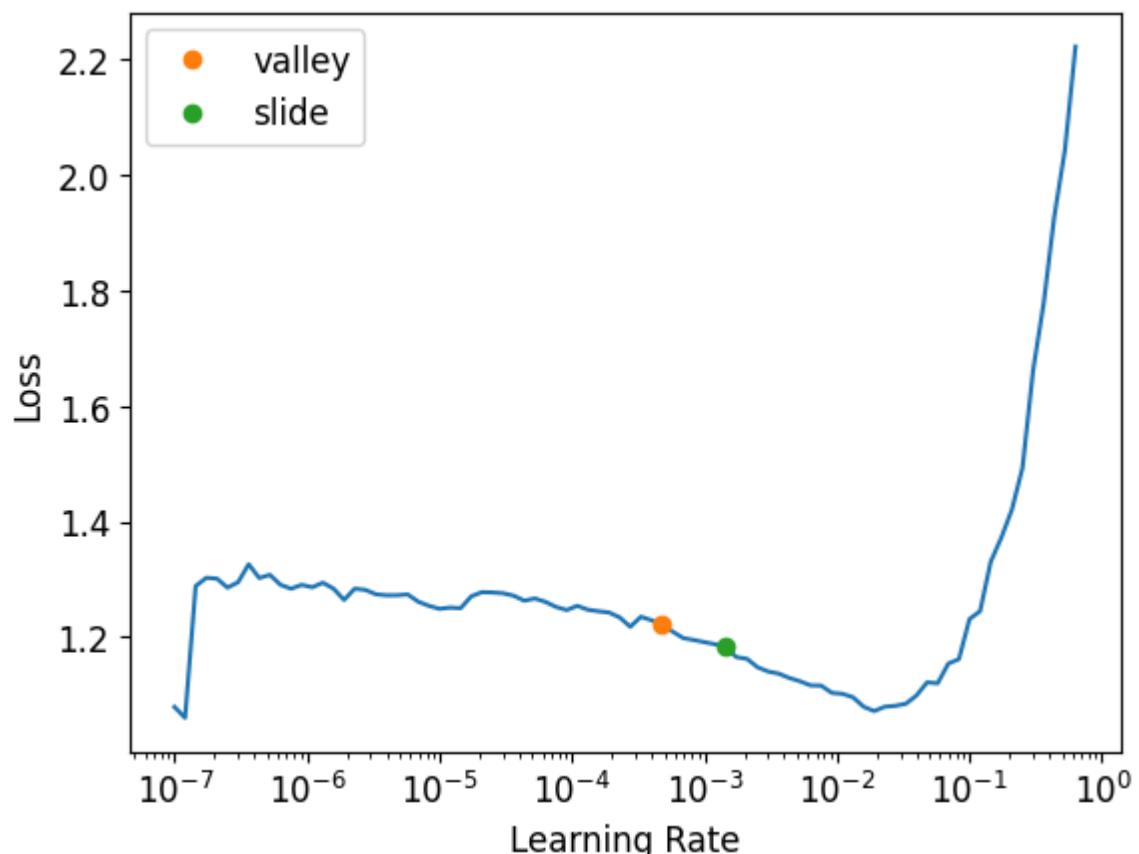
We define `learn6` series for `convnext_small_in22k`.

```
In [ ]: learn6_1, dls6_1, lr6_1 = find_learning_rate_squish(arch2)
```

```
/usr/local/lib/python3.11/dist-packages/timm/models/_factory.py:126:  
UserWarning: Mapping deprecated model name convnext_small_in22k to c  
urrent convnext_small.fb_in22k.  
    model = create_fn()
```

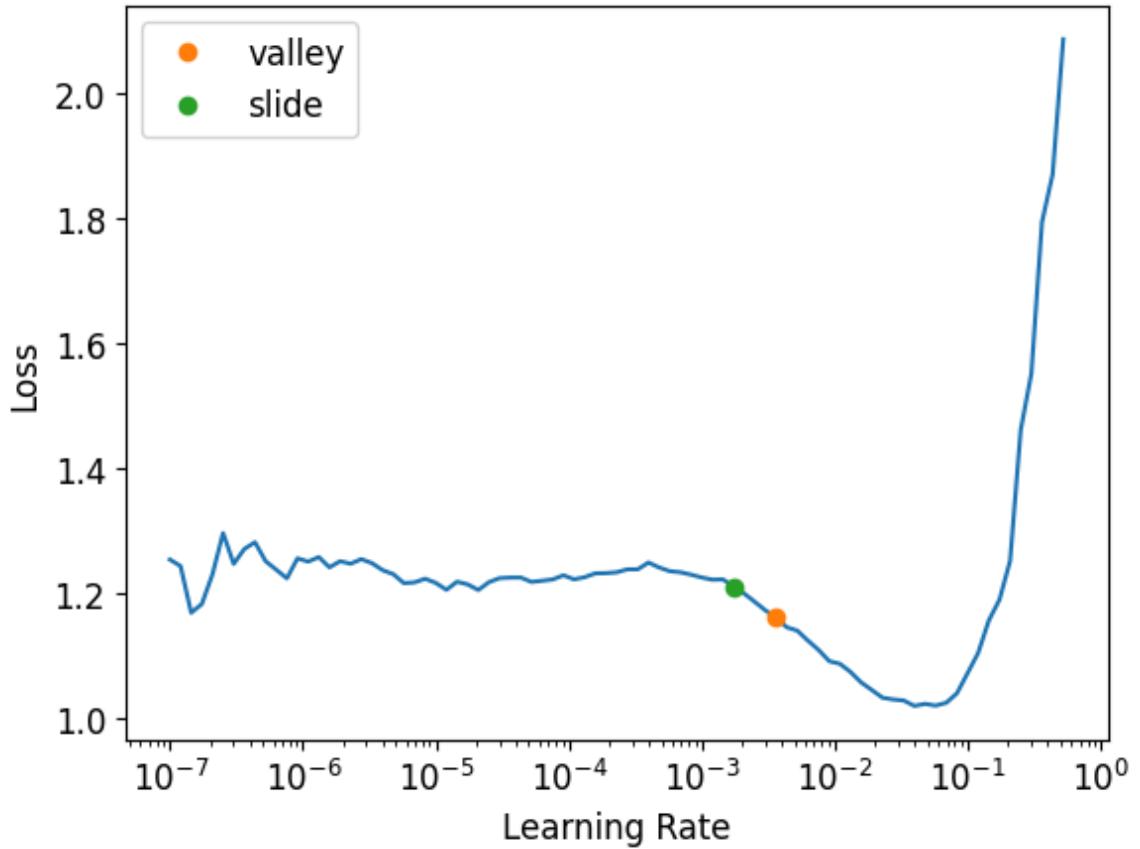
```
model.safetensors:  0% | 0.00/265M [00:00<?, ?B/s]
```

Valley: 0.0004786300996784121; Slide: 0.0014454397605732083



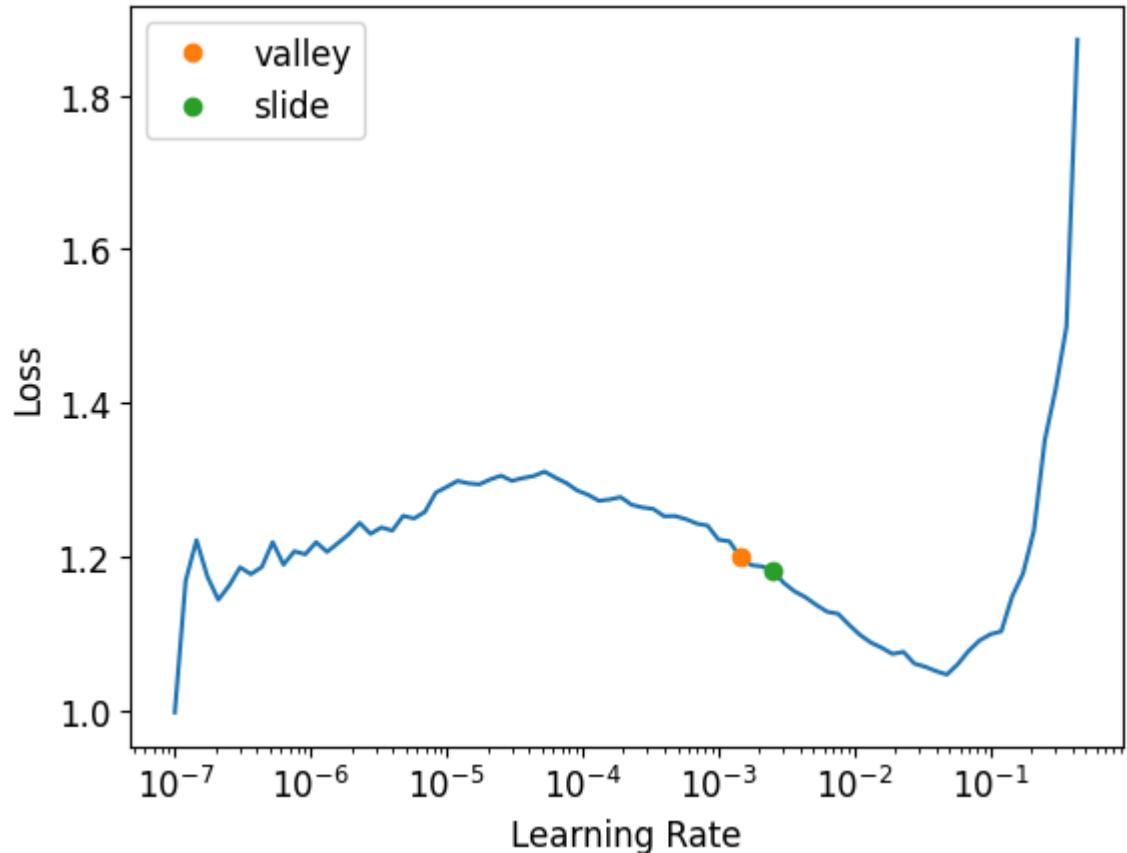
```
In [ ]: learn6_2, dls6_2, lr6_2 = find_learning_rate_zero(arch2)
```

Valley: 0.00363078061491251; Slide: 0.001737800776027143



```
In [ ]: learn6_3, dls6_3, lr6_3 = datablock_method(arch2)
```

Valley: 0.0014454397605732083; Slide: 0.002511886414140463



```
In [ ]: learn6_1.fine_tune(epochs=15, base_lr=lr6_1, freeze_epochs=3)
```

```
name6_1 = 'convnext_small_in22k-squish.csv'  
desc6_1 = 'convnext_small_in22k-squish-training'  
write_to_submission(dls6_1, learn6_1, ss, name6_1, desc6_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.567464	0.374131	0.159265	02:18
1	0.341886	0.270881	0.113357	02:18
2	0.264423	0.213402	0.083001	02:18

epoch	train_loss	valid_loss	error_rate	time
0	0.199608	0.178700	0.067199	04:18
1	0.179453	0.156281	0.056969	04:18
2	0.152659	0.137366	0.049568	04:18
3	0.125231	0.123225	0.045991	04:18
4	0.112584	0.117336	0.042582	04:18
5	0.086008	0.109591	0.038340	04:18
6	0.088324	0.106445	0.037841	04:19
7	0.065904	0.103379	0.034182	04:18
8	0.061373	0.112528	0.035180	04:19
9	0.040158	0.110083	0.034098	04:19
10	0.040789	0.106112	0.031354	04:18
11	0.037736	0.106366	0.031687	04:18
12	0.031828	0.108275	0.030855	04:18
13	0.029213	0.108540	0.031687	04:19
14	0.030997	0.109139	0.032685	04:18

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 99.4kB/s]
```

 convnext_small_in22k-squish.csv	0.70368	0.70089	<input type="checkbox"/>
Complete (after deadline) · 1d ago · convnext_small_in22k-squish-training			

```
In [ ]: learn6_2.fine_tune(epochs=15, base_lr=lr6_2, freeze_epochs=3)
```

```
name6_2 = 'convnext_small_in22k-zero-paddding.csv'  
desc6_2 = 'convnext_small_in22k-zero-paddding-training'  
write_to_submission(dls6_2, learn6_2, ss, name6_2, desc6_2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.401515	0.311395	0.129574	01:33
1	0.258253	0.230559	0.090569	01:33
2	0.190969	0.217325	0.086494	01:33

epoch	train_loss	valid_loss	error_rate	time
0	0.145597	0.185851	0.071856	02:53
1	0.130002	0.132900	0.048403	02:53
2	0.110686	0.141369	0.053144	02:53
3	0.103677	0.111240	0.040170	02:53
4	0.088108	0.111264	0.041750	02:53
5	0.068997	0.106779	0.037924	02:53
6	0.050738	0.096602	0.032269	02:53
7	0.045426	0.109506	0.035679	02:53
8	0.034752	0.130873	0.039920	02:53
9	0.040556	0.097332	0.029691	02:53
10	0.026817	0.108835	0.031520	02:53
11	0.027736	0.108597	0.028110	02:53
12	0.020220	0.104590	0.027944	02:53
13	0.021852	0.109700	0.028443	02:53
14	0.016404	0.105545	0.027112	02:54

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 145kB/s]
```

 convnext_small_in22k-zero-paddding.csv	0.82887	0.83636	<input type="checkbox"/>
Complete (after deadline) · 1d ago · convnext_small_in22k-zero-paddding-training			

```
In [ ]: learn6_3.fine_tune(epochs=15, base_lr=lr6_3, freeze_epochs=3)
```

```
name6_3 = 'convnext_small_in22k-datablock.csv'  
desc6_3 = 'convnext_small_in22k-datablock-training'  
write_to_submission(dls6_3, ss, name6_3, desc6_3)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.405829	0.279188	0.112608	02:04
1	0.222715	0.175908	0.069029	02:04
2	0.170367	0.135946	0.050233	02:04

epoch	train_loss	valid_loss	error_rate	time
0	0.111414	0.112510	0.041500	04:05
1	0.091736	0.104635	0.039421	04:05
2	0.069316	0.090996	0.031104	04:05
3	0.056330	0.094284	0.031520	04:05
4	0.037427	0.099281	0.028526	04:05
5	0.019458	0.114241	0.027362	04:05
6	0.016888	0.116597	0.025699	04:05
7	0.008107	0.129761	0.025283	04:05
8	0.005395	0.121340	0.023120	04:05
9	0.003938	0.132871	0.025200	04:05
10	0.004037	0.138981	0.024534	04:05
11	0.003298	0.134576	0.024118	04:05
12	0.001135	0.134393	0.023037	04:05
13	0.001199	0.135438	0.022372	04:05
14	0.002109	0.135149	0.022954	04:05

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 91.4kB/s]
```

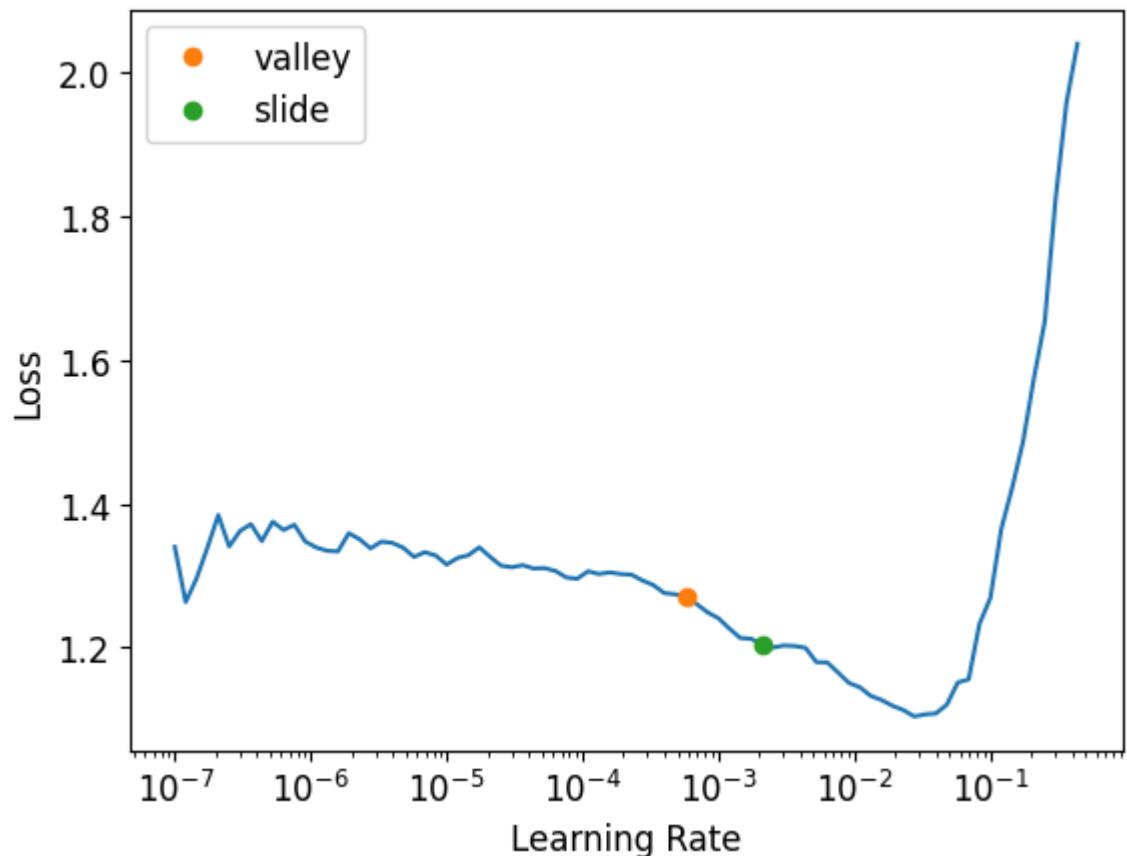
 convnext_small_in22k-datablock.csv	0.71024	0.71646	
Complete (after deadline) · 1d ago · convnext_small_in22k-datablock-training			

We define `learn7` series for `convnextv2_base`.

```
In [ ]: learn7_1, dls7_1, lr7_1 = find_learning_rate_squish(arch3)

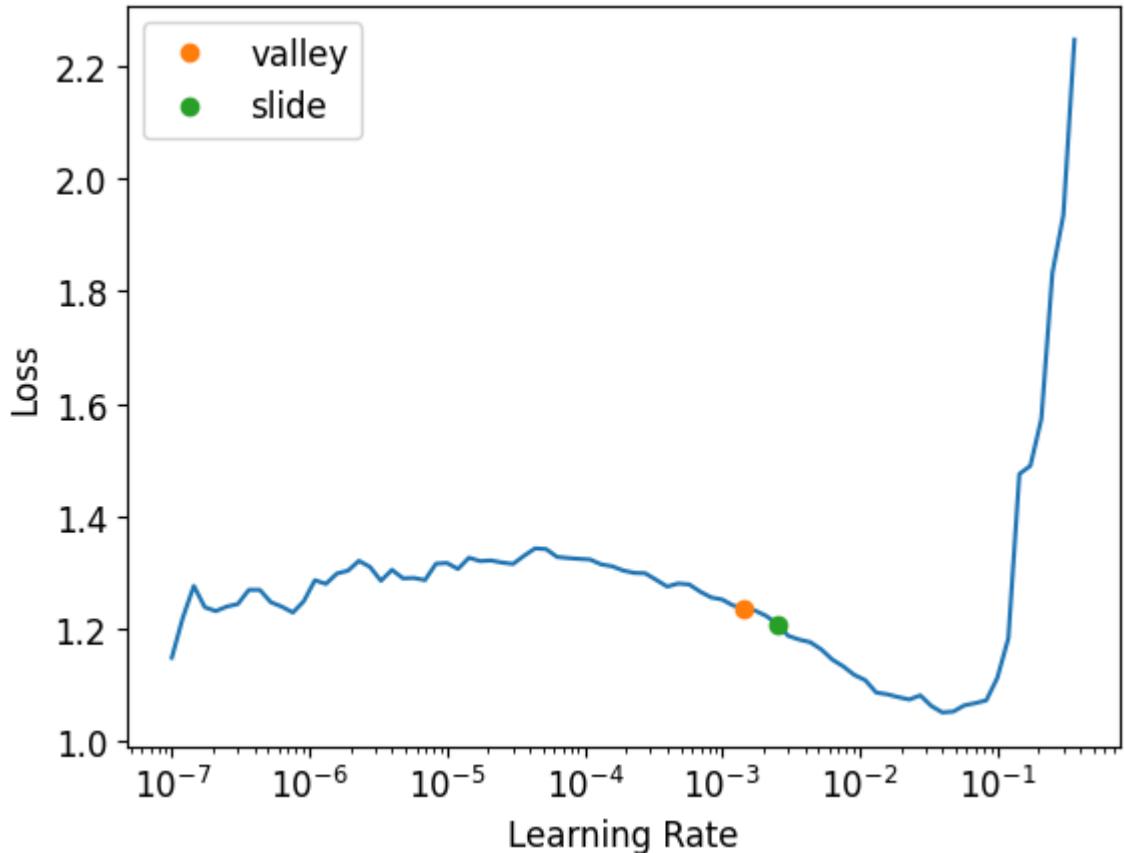
model.safetensors: 0% | 0.00/355M [00:00<?, ?B/s]
```

Valley: 0.0005754399462603033; Slide: 0.0020892962347716093



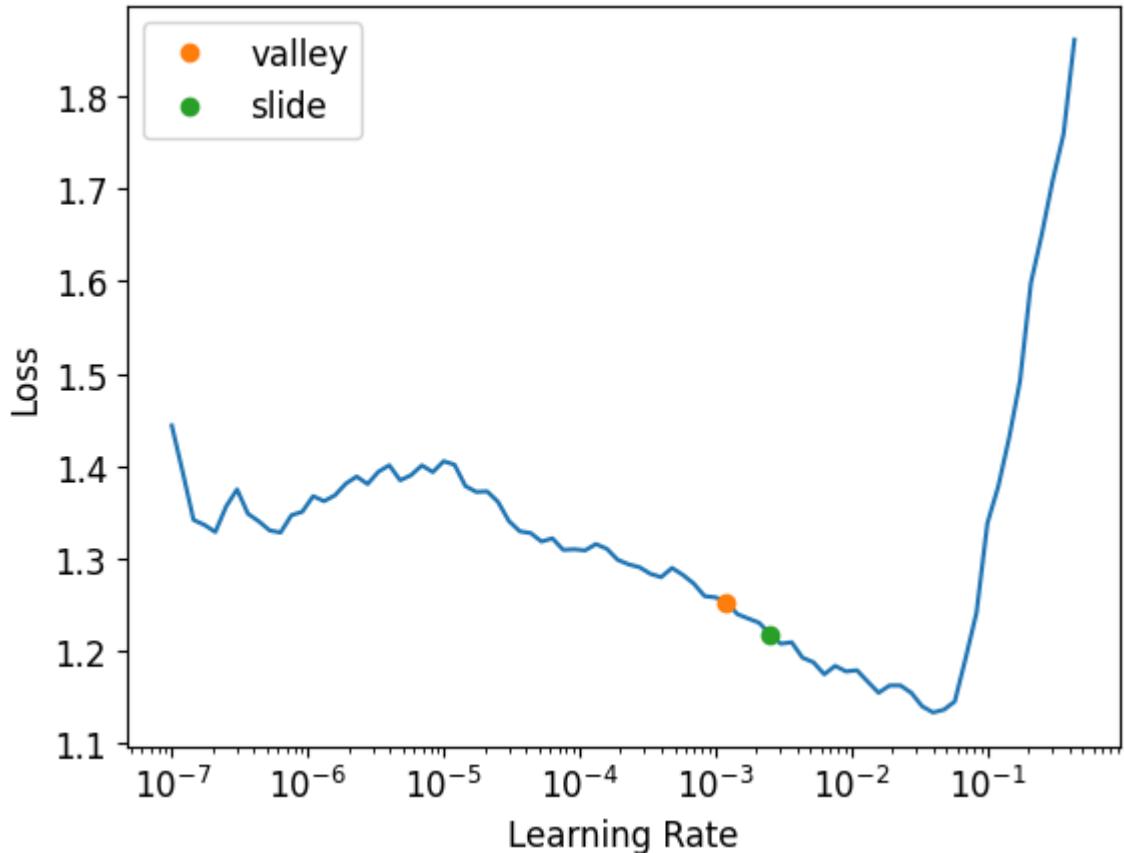
```
In [ ]: learn7_2, dls7_2, lr7_2 = find_learning_rate_zero(arch3)
```

Valley: 0.0014454397605732083; Slide: 0.002511886414140463



```
In [ ]: learn7_3, dls7_3, lr7_3 = datablock_method(arch3)
```

Valley: 0.0012022644514217973; Slide: 0.002511886414140463



```
In [ ]: learn7_1.fine_tune(epochs=15, base_lr=lr7_1, freeze_epochs=3)
```

```
name7_1 = 'convnextv2_base-squish.csv'
desc7_1 = 'convnextv2_base-squish-training'
write_to_submission(dls7_1, learn7_1, ss, name7_1, desc7_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.480265	0.345871	0.147455	04:26
1	0.259014	0.225594	0.090403	04:26
2	0.191206	0.177930	0.068945	04:26

epoch	train_loss	valid_loss	error_rate	time
0	0.157407	0.136933	0.050649	07:21
1	0.115976	0.112399	0.042748	07:20
2	0.092641	0.112530	0.041251	07:20
3	0.070308	0.089185	0.031853	07:20
4	0.049703	0.084374	0.027362	07:21
5	0.043241	0.082699	0.024701	07:21
6	0.030364	0.094124	0.025532	07:21
7	0.024600	0.087814	0.023536	07:21
8	0.020105	0.090931	0.024534	07:21
9	0.010597	0.126441	0.028942	07:21
10	0.008537	0.104984	0.023952	07:21
11	0.005192	0.111456	0.024617	07:21
12	0.004651	0.111473	0.023869	07:20
13	0.003693	0.109863	0.023703	07:21
14	0.004764	0.109301	0.022705	07:20

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.  
    warnings.warn(  
100%|██████████| 52.6k/52.6k [00:00<00:00, 132kB/s]
```

 convnextv2_base-squish.csv	0.74507	0.74494	
Complete (after deadline) · 1d ago · convnextv2_base-squish-training			

The model training process stopped before the model the finishment of `learn7_2` training, this is because my training reached the time limit. The Google stopped the training for me. For the last several models, the time spent on each epoch are all very long. The long training time will consume a lot of compute units. And the `error_rate` not longer reduce as the training epochs increases.

Therefore, For the rest training, I reduced the epochs number for each model. Originally, the training epoch I set was 15, in the following training, I would set the training epoch to 8 epochs.

In []:

Another Training notebook.

Because the former notebook reached the time limit, it stopped before the whole notebook finish training. Here we will do the rest.

In this chapter, I did some of the rest training, because I left my laptop without regulating, the training process stopped before we train the last model with 3 methods.

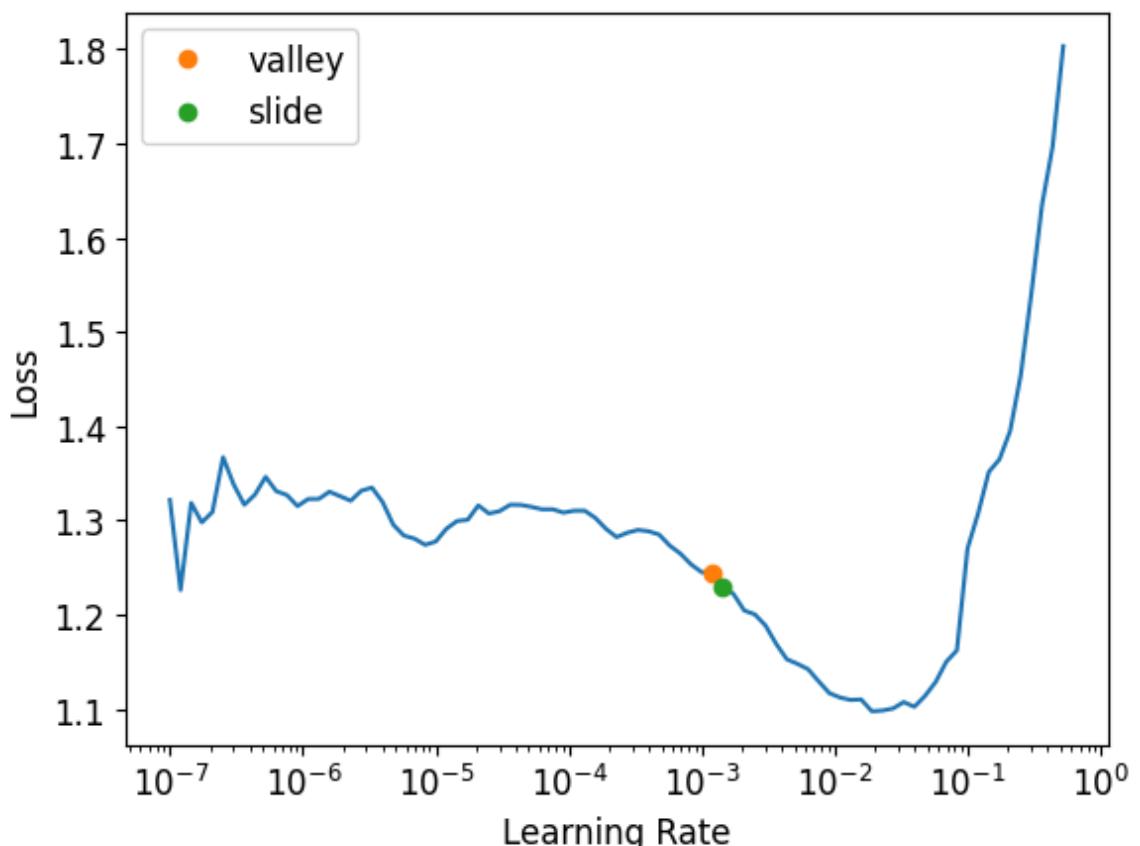
```
In [ ]: arch1 = 'resnet26d'  
arch2 = 'convnext_small_in22k'  
arch3 = 'convnextv2_base'  
arch4 = 'convnextv2_large'
```

Also, find the suitable learning rate for the model. This is the code for training convnext_small_in22k model. In this time I reduced the training epoch for the model training for saving computation resources.

```
In [ ]: learn6_1, dls6_1, lr6_1 = find_learning_rate_squish(arch2)
```

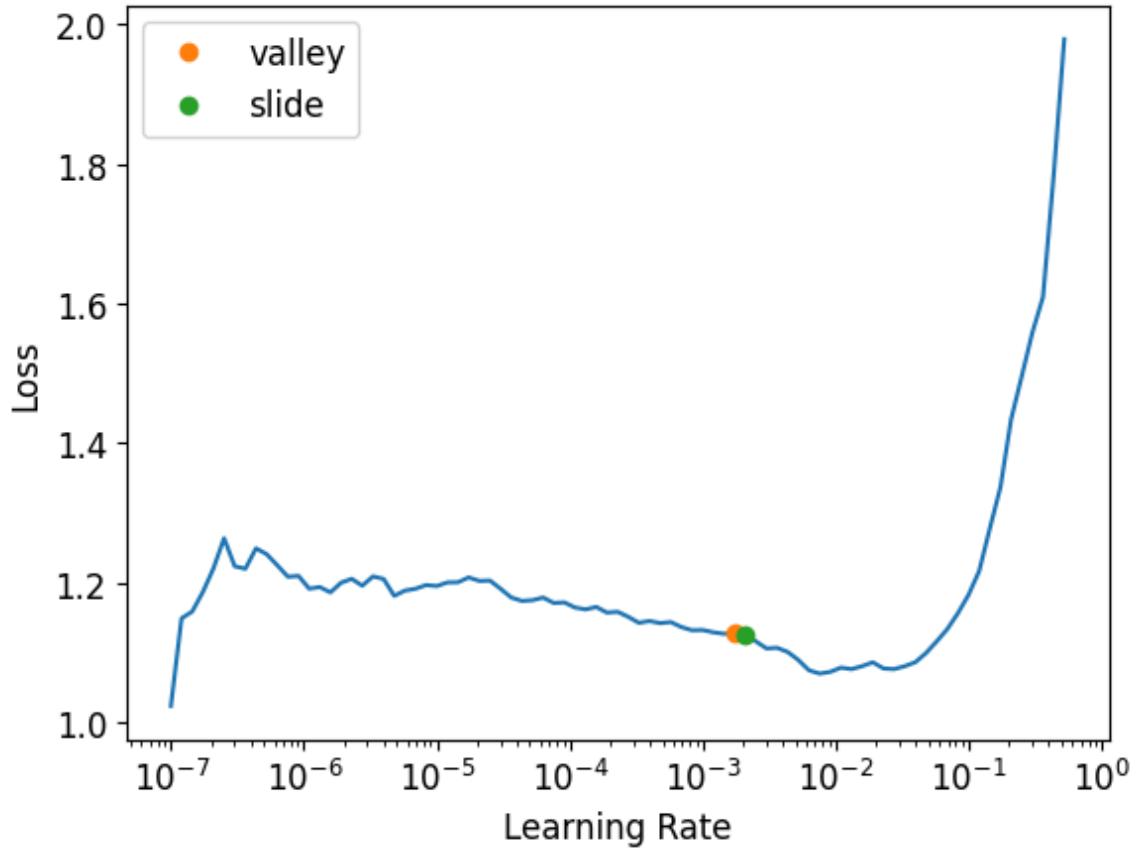
```
/usr/local/lib/python3.11/dist-packages/timm/models/_factory.py:126:  
UserWarning: Mapping deprecated model name convnext_small_in22k to current  
convnext_small.fb_in22k.  
    model = create_fn()  
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.  
py:94: UserWarning:  
The secret `HF_TOKEN` does not exist in your Colab secrets.  
To authenticate with the Hugging Face Hub, create a token in your se  
ttings tab (https://huggingface.co/settings/tokens), set it as secre  
t in your Google Colab and restart your session.  
You will be able to reuse this secret in all of your notebooks.  
Please note that authentication is recommended but still optional to  
access public models or datasets.  
    warnings.warn(  
        model.safetensors: 0% | 0.00/265M [00:00<?, ?B/s]
```

Valley: 0.0012022644514217973; Slide: 0.0014454397605732083



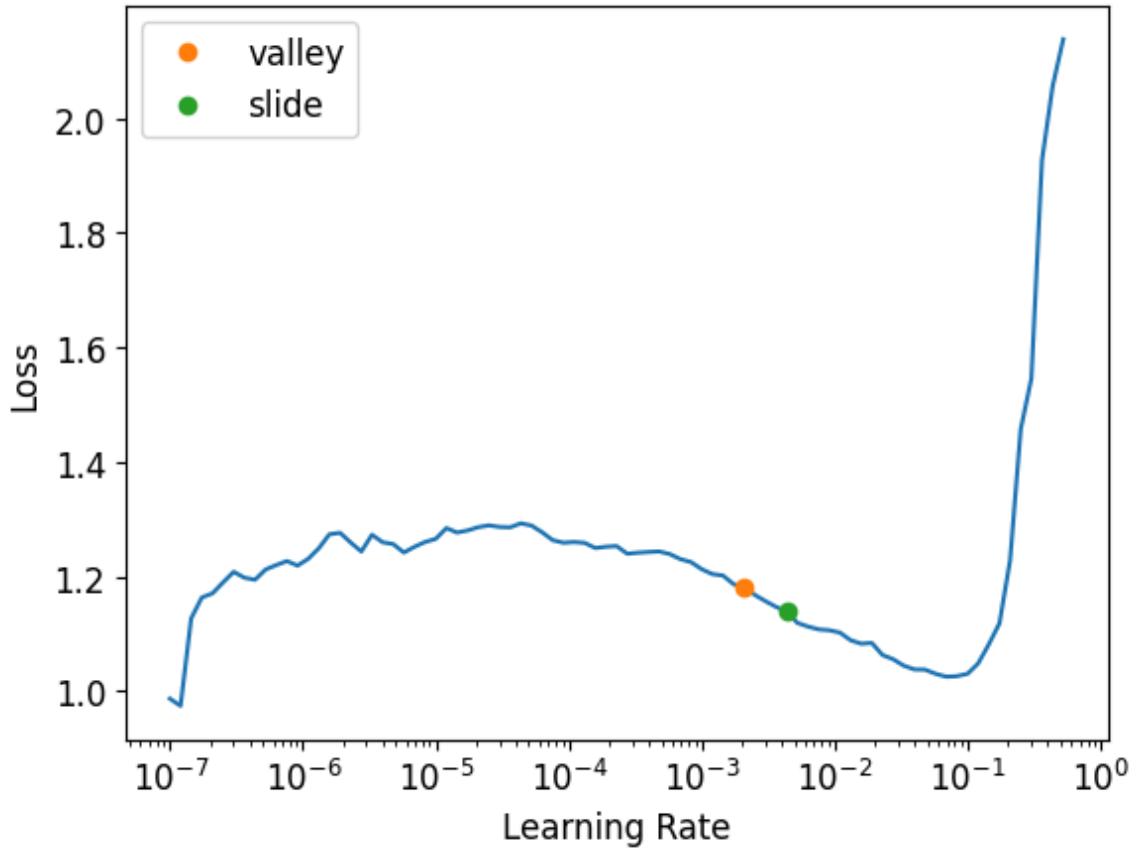
```
In [ ]: learn6_2, dls6_2, lr6_2 = find_learning_rate_zero(arch2)
```

Valley: 0.001737800776027143; Slide: 0.0020892962347716093



```
In [ ]: learn6_3, dls6_3, lr6_3 = datablock_method(arch2)
```

Valley: 0.0020892962347716093; Slide: 0.004365158267319202



```
In [ ]: learn6_1.fine_tune(epochs=8, base_lr=lr6_1, freeze_epochs=3)
```

```
name6_1 = 'convnext_small_in22k-squish.csv'
desc6_1 = 'convnext_small_in22k-squish-training'
write_to_submission(dls6_1, learn6_1, ss, name6_1, desc6_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.566951	0.348538	0.148536	02:17
1	0.330675	0.237809	0.096058	02:17
2	0.247425	0.198140	0.077595	02:17

epoch	train_loss	valid_loss	error_rate	time
0	0.188352	0.156238	0.058300	04:18
1	0.156859	0.131608	0.047904	04:18
2	0.121007	0.117067	0.041168	04:18
3	0.109109	0.104032	0.037259	04:18
4	0.077399	0.107263	0.036926	04:18
5	0.070535	0.105018	0.034265	04:18
6	0.061581	0.096752	0.030273	04:18
7	0.054161	0.099157	0.031104	04:18

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 98.0kB/s]
```

Here is the new training results for the model convnext_small_in22k with 8 epochs per train.



convnext_small_in22k-squish.csv

Complete (after deadline) · 1d ago · convnext_small_in22k-squish-training

0.70721

0.69956



```
In [ ]: learn6_2.fine_tune(epochs=8, base_lr=lr6_2, freeze_epochs=3)
```

```
name6_2 = 'convnext_small_in22k-zero-paddding.csv'
desc6_2 = 'convnext_small_in22k-zero-paddding-training'
write_to_submission(dls6_2, learn6_2, ss, name6_2, desc6_2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.472861	0.328326	0.143463	01:32
1	0.268702	0.239135	0.089904	01:32
2	0.197228	0.210597	0.083832	01:32

epoch	train_loss	valid_loss	error_rate	time
0	0.158629	0.159564	0.062625	02:52
1	0.119857	0.156122	0.061045	02:52
2	0.112708	0.250710	0.095393	02:52
3	0.084930	0.116817	0.044578	02:52
4	0.057315	0.104128	0.038839	02:53
5	0.066832	0.095079	0.033017	02:52
6	0.056033	0.097154	0.034015	02:52
7	0.052565	0.097388	0.034930	02:52

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 97.2kB/s]
```

This is the new train for zero-padding method.



convnext_small_in22k-zero-paddding.csv

Complete (after deadline) · 1d ago · convnext_small_in22k-zero-paddding-training

0.82786

0.83272



```
In [ ]: learn6_3.fine_tune(epochs=8, base_lr=lr6_3, freeze_epochs=3)
```

```
name6_3 = 'convnext_small_in22k-datablock.csv'
desc6_3 = 'convnext_small_in22k-datablock-training'
write_to_submission(dls6_3, learn6_3, ss, name6_3, desc6_3)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.336615	0.232069	0.093147	02:04
1	0.198634	0.176839	0.067781	02:03
2	0.165726	0.123358	0.045160	02:03

epoch	train_loss	valid_loss	error_rate	time
0	0.106842	0.098922	0.035263	04:05
1	0.089103	0.082153	0.028526	04:04
2	0.047845	0.083830	0.026364	04:05
3	0.033856	0.092547	0.025782	04:05
4	0.015033	0.115268	0.026697	04:04
5	0.003439	0.102662	0.021208	04:04
6	0.002310	0.107719	0.021457	04:05
7	0.003057	0.103876	0.020875	04:04

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 104kB/s]
```

This is for the DataBlock method.



convnext_small_in22k-datablock.csv

Complete (after deadline) · 1d ago · convnext_small_in22k-datablock-training

0.71731

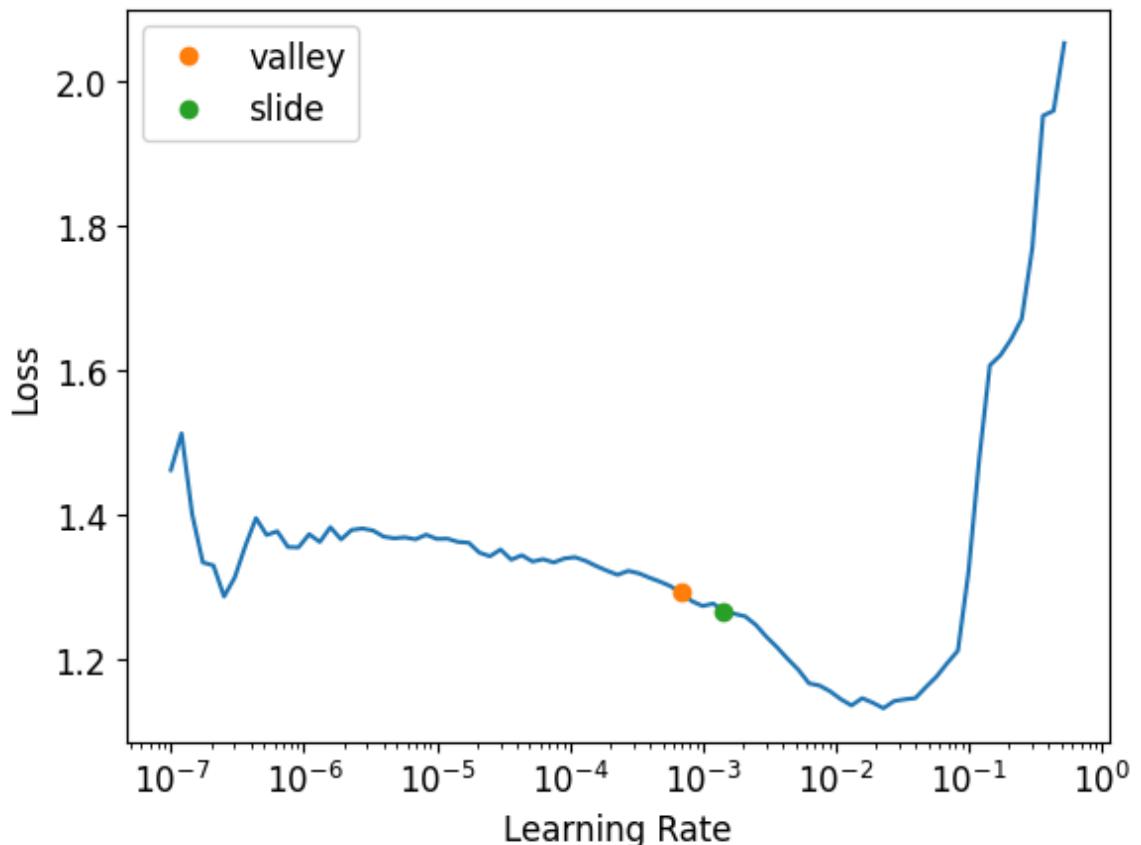
0.73236



```
In [ ]: learn7_1, dls7_1, lr7_1 = find_learning_rate_squish(arch3)
```

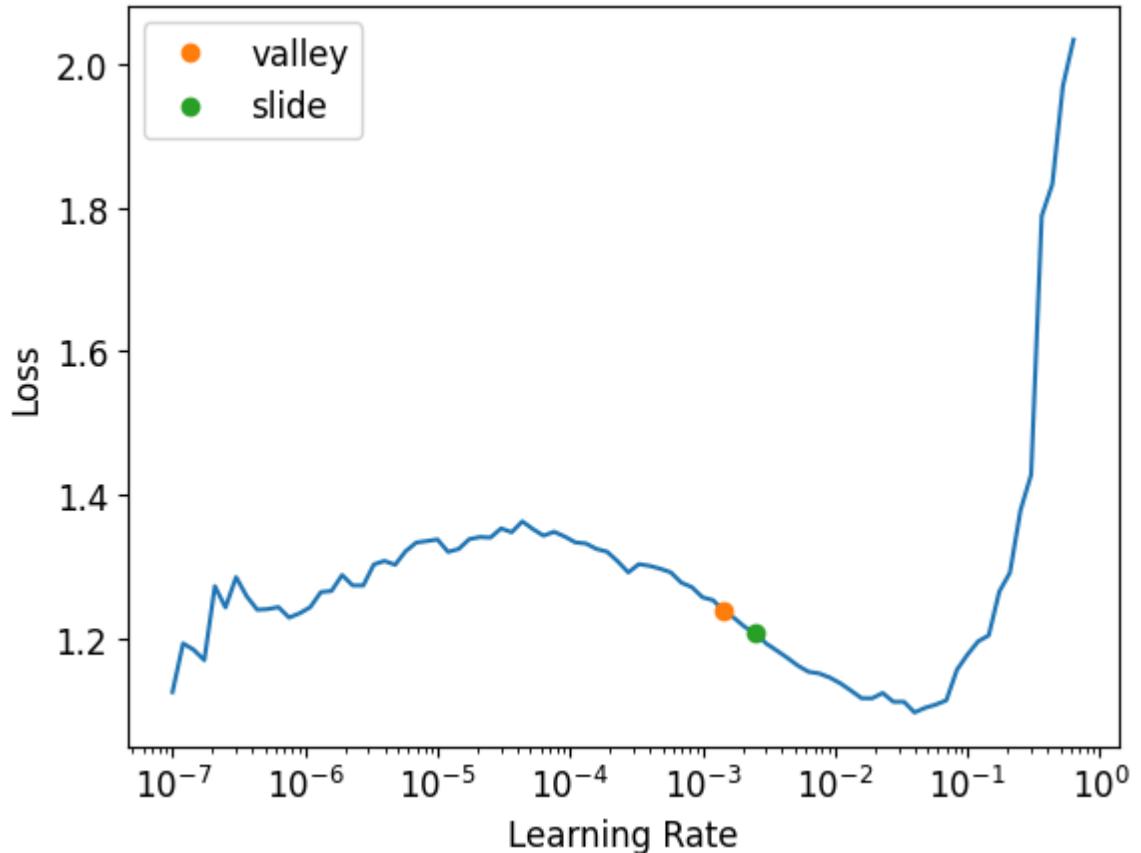
```
model.safetensors: 0% | 0.00/355M [00:00<?, ?B/s]
```

Valley: 0.0006918309954926372; Slide: 0.0014454397605732083



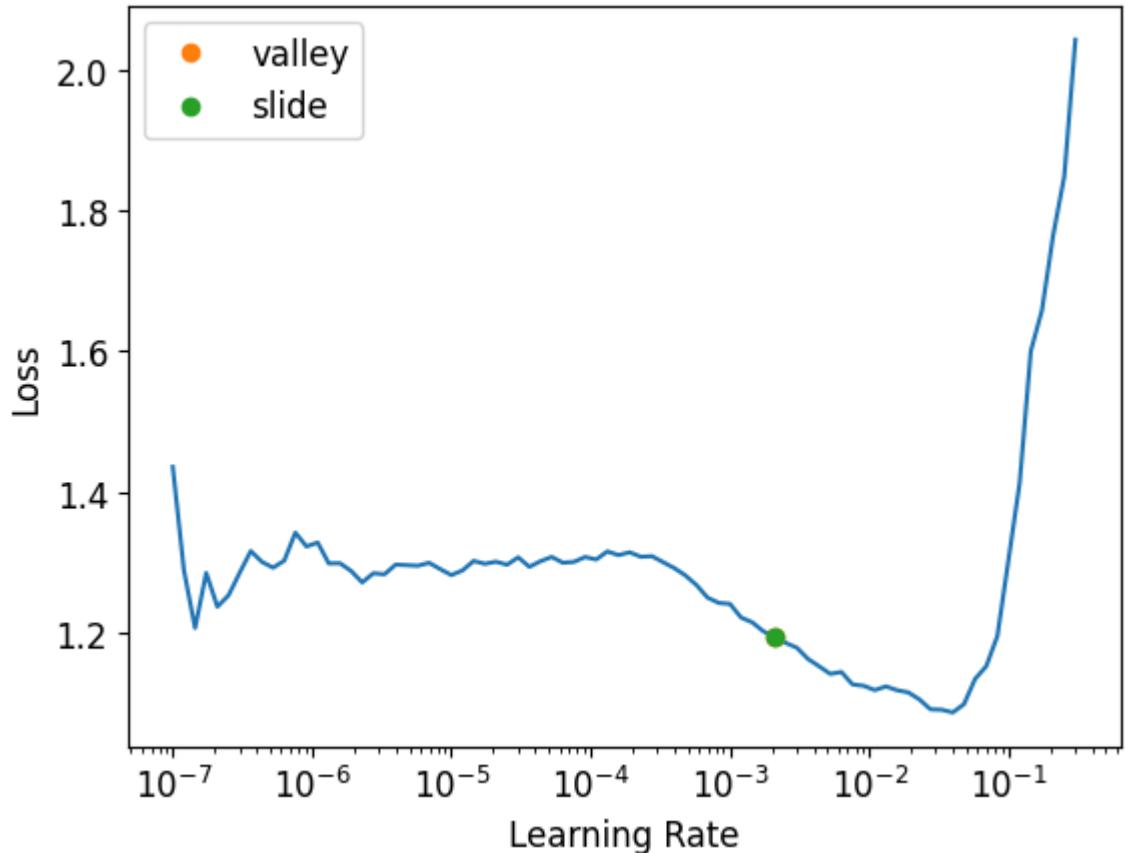
```
In [ ]: learn7_2, dls7_2, lr7_2 = find_learning_rate_zero(arch3)
```

Valley: 0.0014454397605732083; Slide: 0.002511886414140463



```
In [ ]: learn7_3, dls7_3, lr7_3 = datablock_method(arch3)
```

Valley: 0.0020892962347716093; Slide: 0.0020892962347716093



```
In [ ]: learn7_1.fine_tune(epochs=8, base_lr=lr7_1, freeze_epochs=3)
```

```
name7_1 = 'convnextv2_base-squish.csv'
desc7_1 = 'convnextv2_base-squish-training'
write_to_submission(dls7_1, learn7_1, ss, name7_1, desc7_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.515474	0.337744	0.146540	04:25
1	0.286195	0.227642	0.090735	04:25
2	0.206568	0.172369	0.068114	04:25

epoch	train_loss	valid_loss	error_rate	time
0	0.159759	0.129880	0.047239	07:20
1	0.111298	0.102515	0.034182	07:20
2	0.087288	0.092653	0.031687	07:20
3	0.070517	0.080466	0.027112	07:20
4	0.042676	0.084589	0.026530	07:20
5	0.036608	0.078664	0.024202	07:20
6	0.027750	0.077515	0.023287	07:20
7	0.021834	0.084466	0.024035	07:20

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 114kB/s]
```



convnextv2_base-squish.csv

Complete (after deadline) · 1d ago · convnextv2_base-squish-training

0.74659

0.74925



```
In [ ]: learn7_2.fine_tune(epochs=8, base_lr=lr7_2, freeze_epochs=3)
```

```
name7_2 = 'convnextv2_base-zero-paddding.csv'
desc7_2 = 'convnextv2_base-zero-paddding-training'
write_to_submission(dls7_2, learn7_2, ss, name7_2, desc7_2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.423897	0.315564	0.131071	02:41
1	0.247872	0.210025	0.082751	02:42
2	0.173092	0.214463	0.087325	02:42

epoch	train_loss	valid_loss	error_rate	time
0	0.116862	0.191215	0.077595	04:36
1	0.092389	0.179821	0.069694	04:36
2	0.071230	0.104992	0.039172	04:36
3	0.059588	0.104748	0.037176	04:36
4	0.047397	0.141361	0.045576	04:36
5	0.027239	0.099415	0.031354	04:36
6	0.029212	0.133312	0.040253	04:36
7	0.018560	0.110682	0.034597	04:36

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 142kB/s]
```



convnextv2_base-zero-paddding.csv

Complete (after deadline) · 1d ago · convnextv2_base-zero-paddding-training

0.83139

0.84067



```
In [ ]: learn7_3.fine_tune(epochs=8, base_lr=lr7_3, freeze_epochs=3)
```

```
name7_3 = 'convnextv2_base-datablock.csv'
desc7_3 = 'convnextv2_base-datablock-training'
write_to_submission(dls7_3, learn7_3, ss, name7_3, desc7_3)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.423993	0.267454	0.111444	04:11
1	0.205650	0.167674	0.065203	04:11
2	0.151570	0.123914	0.044744	04:11

epoch	train_loss	valid_loss	error_rate	time
0	0.093517	0.087292	0.032102	07:06
1	0.065730	0.075755	0.026114	07:06
2	0.035422	0.075659	0.022455	07:07
3	0.023377	0.081757	0.020709	07:07
4	0.011588	0.092017	0.020792	07:07
5	0.004072	0.091323	0.017964	07:07
6	0.002106	0.095823	0.017881	07:07
7	0.001270	0.094928	0.018047	07:06

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 109kB/s]
```



convnextv2_base-datablock.csv

Complete (after deadline) · 1d ago · convnextv2_base-datablock-training

0.73447

0.75190



Till here is the training process of model `convnext_small_in22k` and model `convnextv2_base` models.

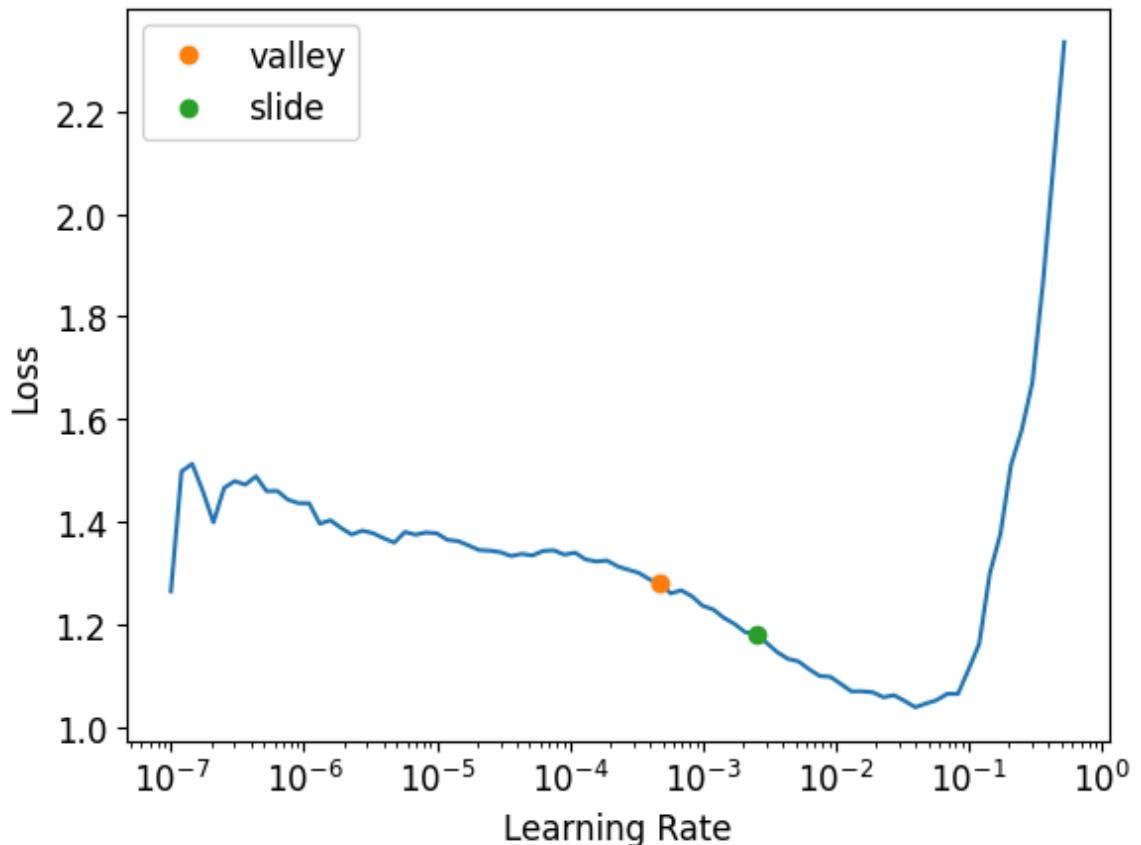
This is last training process for the model convnextv2_large .

```
In [ ]: arch1 = 'resnet26d'  
arch2 = 'convnext_small_in22k'  
arch3 = 'convnextv2_base'  
arch4 = 'convnextv2_large'
```

Find the suitable learning rate.

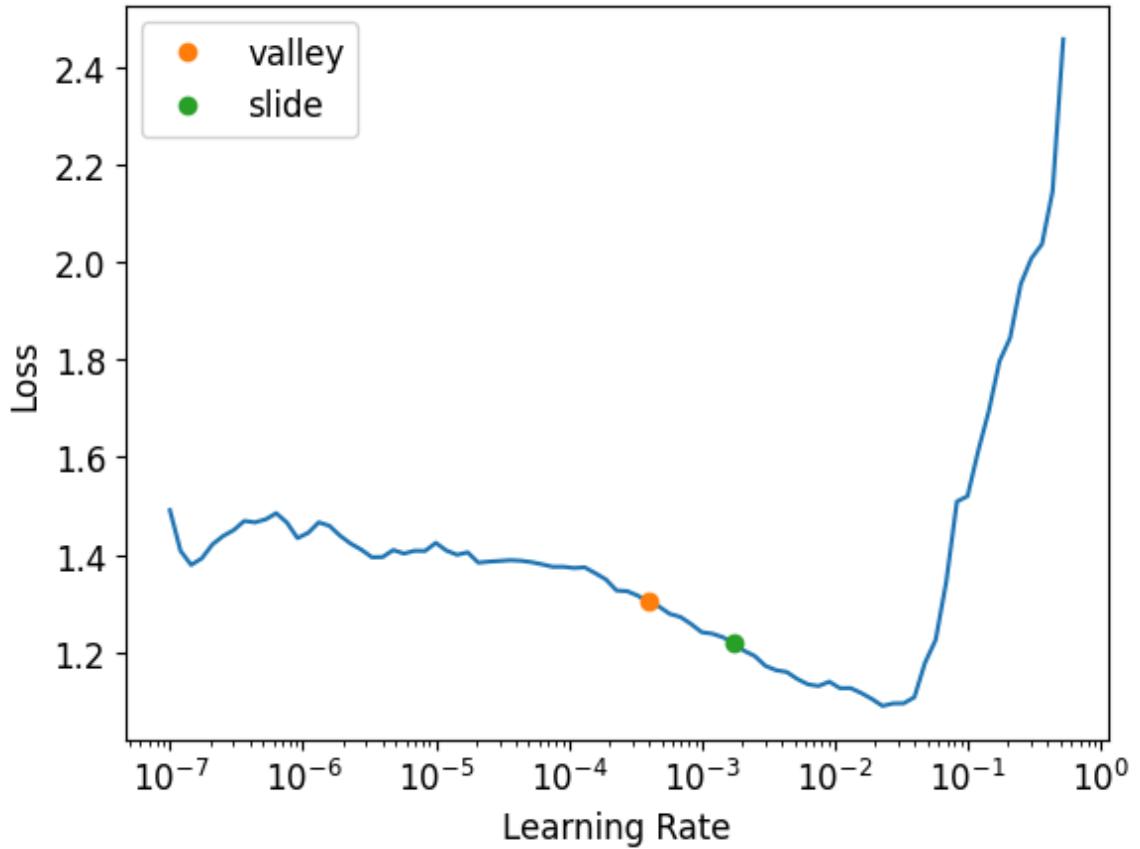
```
In [ ]: learn8_1, dls8_1, lr8_1 = find_learning_rate_squish(arch4)  
  
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:  
The secret `HF_TOKEN` does not exist in your Colab secrets.  
To authenticate with the Hugging Face Hub, create a token in your se  
ttings tab (https://huggingface.co/settings/tokens), set it as secre  
t in your Google Colab and restart your session.  
You will be able to reuse this secret in all of your notebooks.  
Please note that authentication is recommended but still optional to  
access public models or datasets.  
    warnings.warn(  
  
model.safetensors: 0% | 0.00/792M [00:00<?, ?B/s]
```

Valley: 0.0004786300996784121; Slide: 0.002511886414140463



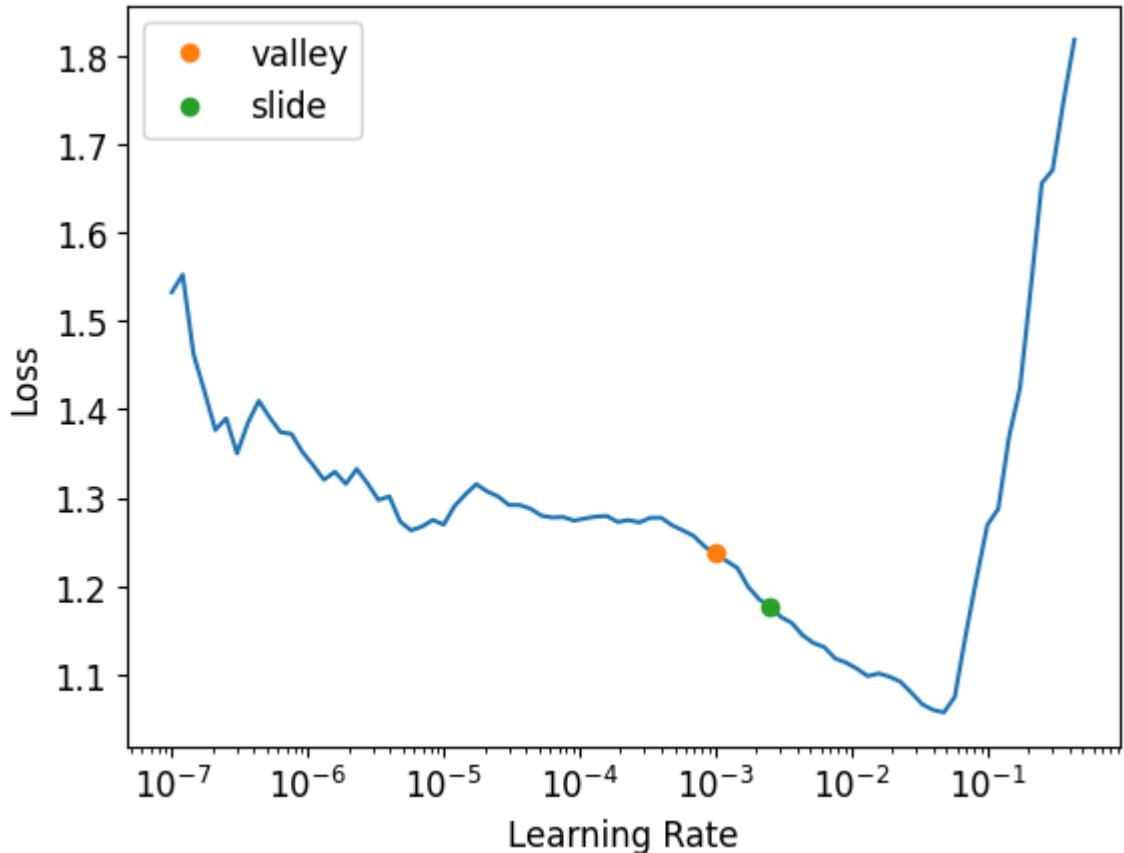
```
In [ ]: learn8_2, dls8_2, lr8_2 = find_learning_rate_zero(arch4)
```

Valley: 0.0003981071640737355; Slide: 0.001737800776027143



```
In [ ]: learn8_3, dls8_3, lr8_3 = datablock_method(arch4)
```

Valley: 0.0010000000474974513; Slide: 0.002511886414140463



```
In [ ]: learn8_1.fine_tune(epochs=8, base_lr=lr8_1, freeze_epochs=3)
```

```
name8_1 = 'convnextv2_large-squish.csv'
desc8_1 = 'convnextv2_large-squish-training'
write_to_submission(dls8_1, learn8_1, ss, name8_1, desc8_1)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.420777	0.288615	0.117598	07:39
1	0.236338	0.219106	0.087325	07:40
2	0.184679	0.151960	0.058217	07:39

epoch	train_loss	valid_loss	error_rate	time
0	0.125778	0.116589	0.044328	12:24
1	0.090141	0.094150	0.034764	12:23
2	0.063058	0.090276	0.033017	12:23
3	0.046714	0.086371	0.029108	12:24
4	0.024357	0.081249	0.024867	12:26
5	0.016696	0.091075	0.024784	12:24
6	0.009559	0.082167	0.023204	12:24
7	0.006884	0.085254	0.023453	12:24

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 72.5kB/s]
```



convnextv2_large-squish.csv

Complete (after deadline) · 19h ago · convnextv2_large-squish-training

0.74154

0.73964



```
In [ ]: learn8_2.fine_tune(epochs=8, base_lr=lr8_2, freeze_epochs=3)
```

```
name8_2 = 'convnextv2_large-zero-paddding.csv'
desc8_2 = 'convnextv2_large-zero-paddding-training'
write_to_submission(dls8_2, ss, name8_2, desc8_2)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.426101	0.346317	0.140552	04:38
1	0.230672	0.265049	0.110279	04:38
2	0.189891	0.175760	0.068613	04:38

epoch	train_loss	valid_loss	error_rate	time
0	0.131109	0.169634	0.068862	07:35
1	0.107552	0.138882	0.054308	07:35
2	0.082938	0.171312	0.066035	07:35
3	0.057037	0.111851	0.042083	07:35
4	0.035673	0.093523	0.033516	07:35
5	0.040301	0.090487	0.032768	07:35
6	0.032560	0.097334	0.033433	07:35
7	0.030205	0.103873	0.035346	07:35

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 119kB/s]
```



convnextv2_large-zero-paddding.csv

Complete (after deadline) · 18h ago · convnextv2_large-zero-paddding-training

0.83139

0.84233



```
In [ ]: learn8_3.fine_tune(epochs=8, base_lr=lr8_3, freeze_epochs=3)
```

```
name8_3 = 'convnextv2_large-datablock.csv'
desc8_3 = 'convnextv2_large-datablock-training'
write_to_submission(dls8_3, ss, name8_3, desc8_3)
```

epoch	train_loss	valid_loss	error_rate	time
0	0.358713	0.231115	0.092814	07:26
1	0.196702	0.153118	0.059049	07:26
2	0.153348	0.114552	0.043413	07:26

epoch	train_loss	valid_loss	error_rate	time
0	0.089728	0.081597	0.029358	12:10
1	0.064865	0.068018	0.023536	12:10
2	0.030225	0.070671	0.020709	12:10
3	0.016214	0.085449	0.020875	12:10
4	0.006741	0.087071	0.018047	12:10
5	0.002533	0.085952	0.016883	12:10
6	0.001383	0.088706	0.017465	12:10
7	0.001994	0.087324	0.017132	12:10

```
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
/usr/local/lib/python3.11/dist-packages/PIL/Image.py:3442: DecompressionBombWarning: Image size (99991727 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS attack.
    warnings.warn(
100%|██████████| 52.6k/52.6k [00:00<00:00, 107kB/s]
```



convnextv2_large-datablock.csv

Complete (after deadline) · 16h ago · convnextv2_large-datablock-training

0.73296

0.73666



The content above is the whole training process.

Revision about this assignment.

In this assignment, I tried to train models to do the task to determine whether an image is a real image or generated by AI. I tried several methods on this task, combined with the method I learned from the assignment 2 and assignment 3. The result of the training is shown in the following Table 2.1.

After I browsed all the results of this task, I find that the model with the best performance is the model trained with ‘Resnet18’ and the zero-padding method, whose private score reaches 0.90863 and the public score reaches 0.91321. The second highest is the model ‘Resnet34’ with zero-padding method as well, both private score and public score for this method over 0.90. The rest models cannot have such a performance like this.

And among all the trained models, I find that for the training images, if the model used the zero-padding method, the performances are all better than other methods. This might because after the processing of this method, the useful content is kept more than the squish method, for the squish method, some information of the images will be filtered in our image pro-processing step.

To the squish and Datablock method, I find that their performance is almost the same, this is because these two methods processing images in the same way, of course this will bring the similar result. Originally, I think the label ‘AI’ and ‘REAL’ can be divided into two different categories, therefore we can train a model like what we did in assignment 2.

To the final performance of the models, we find that the error rate of the models is very low at the last several steps, and after we received the low error rate results, we keep training this model will bring a problem of ‘overfitting’ [6]. That is, the model performs well in the given training

dataset, however, when the model faces new dataset, the performance will decrease sharply. The best model we trained in this session is ‘resnet18’ and ‘resnet34’ with zero-padding, by watching the performance in training dataset and the performance in the validation dataset, we find that the error rate is almost same as its training performance, this illustrates that these two models does not has a problem of overfitting.

Possible Future works

There is an important step before we start the model training, that is resize_image, we resized the images to 400x400, then feed the images to the model. However, when we are testing the images in the validation dataset, we do not perform the same process to the validation dataset. So, in the future, I might write a function to resize the validation dataset images before we test the model.

Secondly, we notice that the trend of the error rate, normally for the last several epochs, we might adjust different training epochs for the model. To make sure there will not have problem of overfitting.

Reference

[1] <https://amit-manbansho1.medium.com/have-you-ever-encountered-that-your-keras-image-classification-model-is-training-very-slow-during-962f1c1bb95f>

[2] <https://github.com/fastai/course22/blob/master/o8-first-steps-road-to-the-top-part-1.ipynb>

[3] <https://github.com/fastai/course22/blob/master/o9-small-models-road-to-the-top-part-2.ipynb>

[4] <https://github.com/fastai/course22/blob/master/o10-scaling-up-road-to-the-top-part-3.ipynb>

[5] <https://fastai1.fast.ai/vision.models.html>

[6] <https://developers.google.com/machine-learning/crash-course/overfitting/overfitting#:~:text=Overfitting%20means%20creating%20a%20model,worthless%20in%20the%20real%20world.>