

## RTX 3090

In [1]:

```
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=16)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(4, 1e-2)
```

100.00% [144441344/144440600

00:04&lt;00:00]

100.00% [105070592/105067061 00:03&lt;00:00]

epoch	train_loss	valid_loss	accuracy	time
0	0.470113	0.413801	0.813360	02:16

epoch	train_loss	valid_loss	accuracy	time
0	0.325192	0.233826	0.907080	03:51
1	0.236041	0.264473	0.894120	03:52
2	0.200160	0.194080	0.924960	03:51
3	0.147917	0.184875	0.932120	03:51

In [1]:

```
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=64)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(4, 1e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.458788	0.393556	0.823200	00:43

epoch	train_loss	valid_loss	accuracy	time
0	0.320155	0.261853	0.890760	01:17
1	0.239075	0.207978	0.915560	01:18
2	0.188774	0.186548	0.927800	01:17
3	0.155016	0.189232	0.928600	01:17

In [1]:

```
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=128)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(4, 1e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.466341	0.384222	0.828680	00:39

epoch	train_loss	valid_loss	accuracy	time
-------	------------	------------	----------	------

epoch	train_loss	valid_loss	accuracy	time
0	0.333969	0.247664	0.899160	01:06
1	0.254776	0.215942	0.912880	01:06
2	0.194134	0.189241	0.928120	01:06
3	0.163552	0.201685	0.925840	01:06

In [1]:

```
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=176)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(4, 4e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.466490	0.403492	0.818480	00:34
epoch	train_loss	valid_loss	accuracy	time
0	0.293550	0.328505	0.872360	01:01
1	0.223114	0.217827	0.919520	01:00
2	0.168444	0.186817	0.928080	01:00
3	0.112355	0.194950	0.930440	01:00

In [1]:

```
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=176)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(4, 8e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.475971	0.399147	0.816720	00:34
epoch	train_loss	valid_loss	accuracy	time
0	0.295365	0.233165	0.902720	01:00
1	0.235020	0.205920	0.923160	01:00
2	0.159399	0.206095	0.924360	01:00
3	0.090898	0.205748	0.929600	01:00

In [1]:

```
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=256)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(4, 8e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.479962	0.419820	0.804480	00:17
epoch	train_loss	valid_loss	accuracy	time

epoch	train_loss	valid_loss	accuracy	time
0	0.294295	0.344832	0.824800	00:29
1	0.233454	0.196607	0.923000	00:29
2	0.160799	0.199417	0.927160	00:29
3	0.095754	0.205702	0.933040	00:29

In [1]:

```
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=128*3)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(4, 8e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.479911	0.402575	0.823200	00:17

epoch	train_loss	valid_loss	accuracy	time
0	0.309874	0.231644	0.906440	00:28
1	0.241561	0.263826	0.902720	00:28
2	0.185477	0.203360	0.921720	00:28
3	0.121835	0.207216	0.930000	00:28

In [3]:

```
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=128*3)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(4, 2*8e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.489069	0.513185	0.755680	00:17

epoch	train_loss	valid_loss	accuracy	time
0	0.302114	0.266184	0.885560	00:28
1	0.244610	0.210818	0.921680	00:28
2	0.177167	0.304169	0.901960	00:28
3	0.113023	0.209661	0.925560	00:28

In [4]:

```
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=128*3)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(6, 2*8e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.494179	0.700958	0.650120	00:17

epoch	train_loss	valid_loss	accuracy	time
-------	------------	------------	----------	------

epoch	train_loss	valid_loss	accuracy	time
0	0.306264	0.332416	0.862960	00:28
1	0.255994	0.752627	0.783640	00:28
2	0.205133	0.330665	0.884640	00:28
3	0.142712	0.231974	0.918480	00:28
4	0.080805	0.333080	0.907240	00:28
5	0.044340	0.293192	0.925400	00:28

In [1]:

```

from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=128*3)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy).
learn.fine_tune(6, 8e-2)

```

epoch	train_loss	valid_loss	accuracy	time
0	0.484444	0.480523	0.773800	00:17

epoch	train_loss	valid_loss	accuracy	time
0	0.305955	0.257405	0.907760	00:28
1	0.256423	0.290773	0.870400	00:28
2	0.201120	0.210381	0.915800	00:28
3	0.142036	0.234370	0.919120	00:28
4	0.090502	0.213680	0.927520	00:28
5	0.054446	0.240286	0.930240	00:28

In [1]:

```

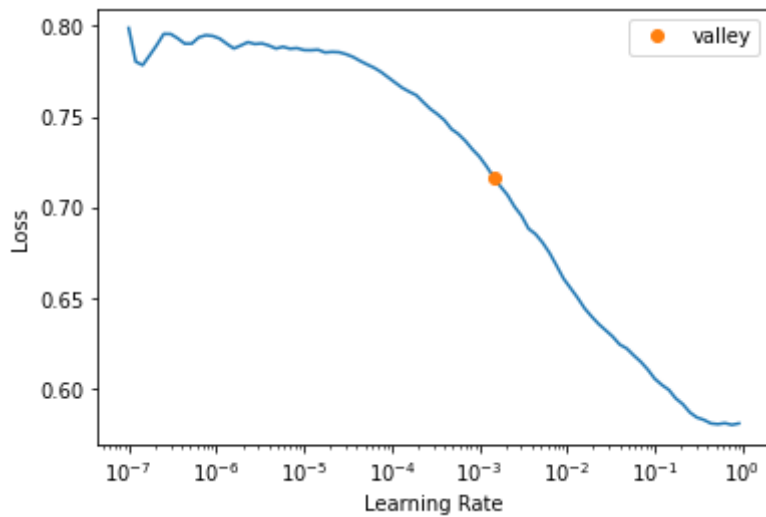
from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=128*3)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy).
learn.lr_find()

```

/opt/conda/lib/python3.8/site-packages/fastai/callback/schedule.py:272: UserWarning: color is redundantly defined by the 'color' keyword argument and the fmt string "ro" (-> color='r'). The keyword argument will take precedence.

```
ax.plot(val, idx, 'ro', label=nm, c=color)
```

Out[1]: SuggestedLRs(valley=0.0014454397605732083)



In [1]:

```

from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=128*3)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(6, 1.4e-3)

```

epoch	train_loss	valid_loss	accuracy	time
0	0.530485	0.392364	0.825800	00:17
epoch	train_loss	valid_loss	accuracy	time
0	0.423949	0.334248	0.855520	00:28
1	0.380431	0.294414	0.874480	00:28
2	0.345084	0.273871	0.886920	00:28
3	0.314887	0.262742	0.893080	00:28
4	0.296610	0.259303	0.894000	00:28
5	0.287854	0.254813	0.896840	00:28

In [1]:

```

from fastai.text.all import *
dls = TextDataLoaders.from_folder(untar_data(URLs.IMDB), valid='test', bs=128*3)
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)
learn.fine_tune(6, 1e-2)

```

epoch	train_loss	valid_loss	accuracy	time
0	0.482049	0.410961	0.814440	00:17
epoch	train_loss	valid_loss	accuracy	time
0	0.358651	0.295170	0.877600	00:28
1	0.305715	0.236679	0.906280	00:28
2	0.257482	0.208283	0.916960	00:28
3	0.213891	0.205402	0.918880	00:28
4	0.177140	0.203323	0.923040	00:28

<b>epoch</b>	<b>train_loss</b>	<b>valid_loss</b>	<b>accuracy</b>	<b>time</b>
5	0.154081	0.211795	0.922560	00:28

In [ ]: