FastAI (neural networks)

DS Development Presentations

Peter Nicholas S. Onglao | MNL

Contents

- Introduction to FastAI
- Techniques being employed by FastAI
- Examples
  - Sentiment analysis (NLP)
  - Image classification (computer vision)

1

Introduction to FastAI

Python package for neural networks

FastAI

- Slogan:
  - “Making neural nets uncool again”
- Mission:
  - To make deep learning more accessible and easier to use
  - Code first approach
    - Tutorials start with implementation, and then learning the specifics come after
- What it does:
  - Simplifies training fast and accurate neural net using modern best practices

FastAI

- Models types/Fast AI Modules:
  - Vision
  - Text
  - Tabular
  - Collab (Collaborative filtering)
    - Recommendation systems; giving suggestions to users based on the likes/dislikes of similar users
    - E.g. Netflix recommendations

FastAI

- Advantages:
  - Very easy to use
  - Employs modern techniques out-of-the-box
  - Has free lessons online, with sample notebooks
  - Has an active forum where the developers frequently reply

FastAI

- Disadvantages:
  - All the developers use Linux
    - Using Windows/MacOS is hard
  - It’s relatively new and unstable
    - There are multiple fastai versions
      - Hard to keep up to date with packages/dependencies
    - Website has some outdated sections

2

Techniques Employed

Advanced deep learning out-of-the-box

Contents

- The One Cycle Policy
- Fine-tuning pretrained models

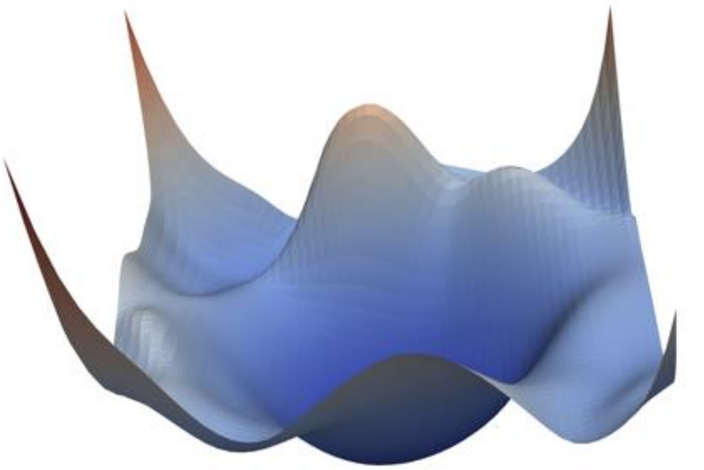
The One Cycle Policy

- Proposed in a series of papers by Leslie N. Smith
- For training very quickly (called *superconvergence*)
- Essentially, it's a way of finding good learning rates

Jeremy Jordan

The One Cycle Policy

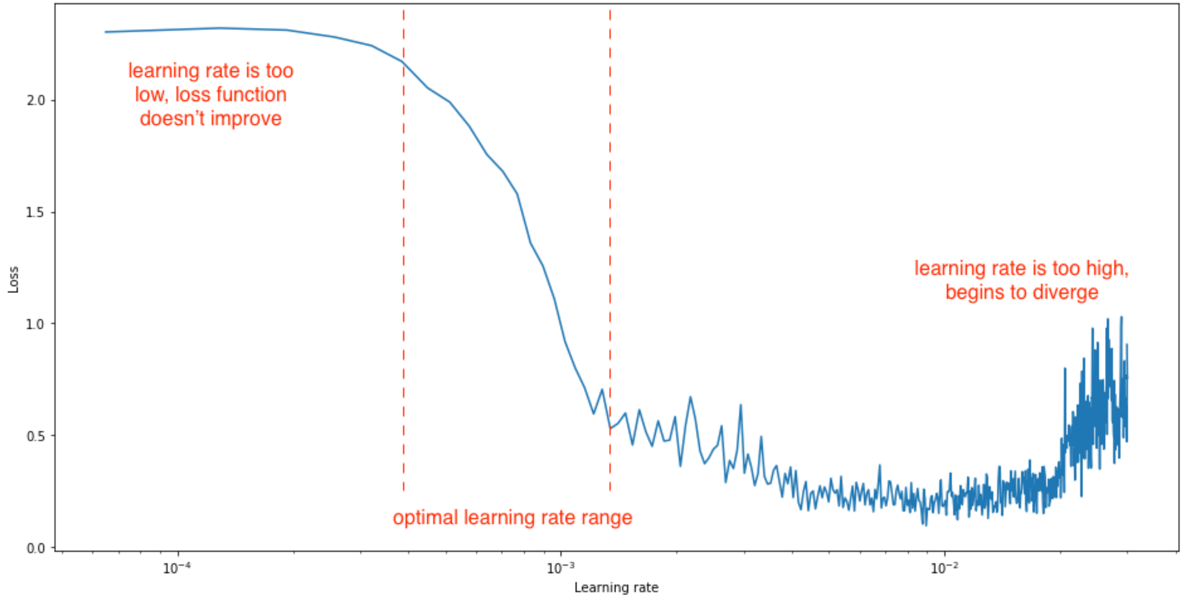
- Minimizing the loss (visualized below) is complicated
- It’s dependent on your architecture and your dataset



Jeremy Jordan

The One Cycle Policy

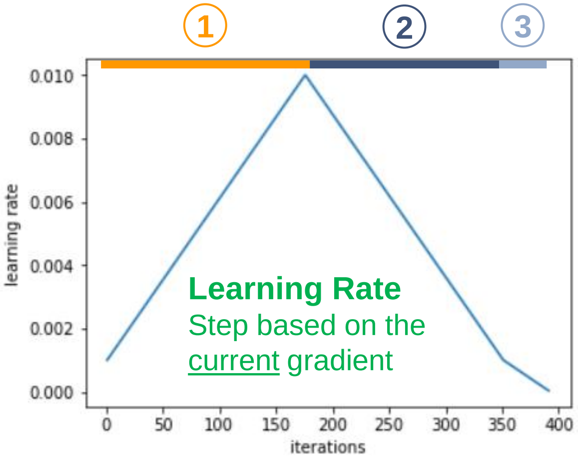
- We need a systemic approach to find the optimal learning rate

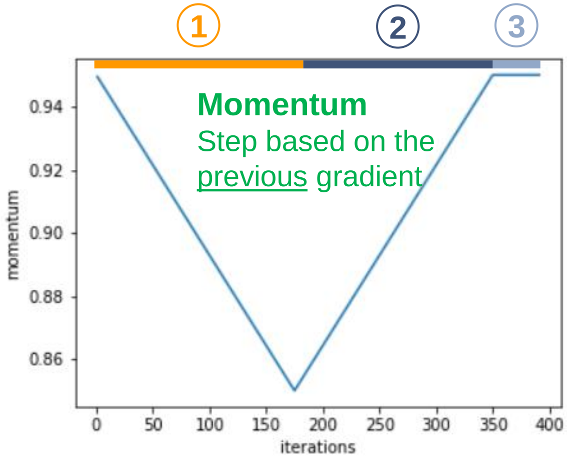


Jeremy Jordan

The One Cycle Policy

- The One Cycle Policy
  1. Increase learning rate and decrease momentum
  2. Decrease learning rate and increase momentum
  3. Diminish learning rate and keep momentum steady





[Sylvain Gugger]

Fine-tuning pretrained models

- FastAI has a variety of pretrained models available
  - Vision – pretrained on ImageNet
    - ResNet
    - SqueezeNet
    - DenseNet
    - VGG
    - AlexNet
    - EfficientNet (new, from Google)
  - Text
    - AWD LSTM (*Average SGD Weight-Dropped LSTM*)
    - Transformer

Fine-tuning pretrained models

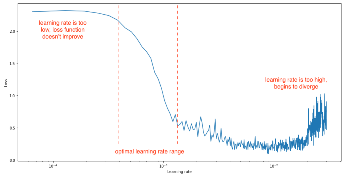
- For NLP, it's not as simple as using pretrained word embeddings
  - Word embeddings: convert a word to a vector
- FastAI follows ULMFiT, *Universal Language Model Fine-tuning for Text Classification*
  - Goes beyond word embeddings; it uses the embeddings to get representations of full sentences/units of your data
- Framework:
  - Preprocess text data
  - Create a language model to encode your text
  - Create a model (e.g. classifier) which uses the encoder you made

Modeling Flowchart

- Basic flowchart of how to model:

Preprocess /clean the data → Create **DataBunch** → **Initial fit** – fit last layer of pretrained model

→ **Unfreeze** some/all layers and run **LR finder** → **Tune** – fit some/all layers of the model → **Interpret** model results



3

Examples

Sentiment Analysis and Image Classification

Sentiment Analysis

Data: Amazon Review

The **classification/sentiment analysis task**: given an Amazon review (review headline + review body), predict the star rating from 1-5.

```
In [2]: df_amazon = pd.read_csv('data/text/amazon_reviews_us_Wireless_v1_00.tsv.gz',
                                sep='\t',
                                compression='gzip',
                                usecols=['review_headline','review_body','star_rating'],
                                error_bad_lines=False,
                                warn_bad_lines=False)
        df_amazon = df_amazon.dropna()
        df_amazon = df_amazon.sample(100000, random_state=2020)
```

Data file

Columns to use

Take a 100,000 sample (orig. size: ~9M)

Modeling – Preprocess data (1 – Preprocess)

Remove punctuation

Remove whitespace, '-', and '/' characters

Lowercase

Concat headline and body

Convert ratings to integer

In [3]:
```
REPLACE_NO_SPACE = re.compile("[.;:!\'?,\"()\[\]]")
REPLACE_WITH_SPACE = re.compile("(<br\s*/><br\s*/>)|(\-)|(\/)")

def preprocess_reviews(reviews):
    reviews = [REPLACE_NO_SPACE.sub("", line.lower()) for line in reviews]
    reviews = [REPLACE_WITH_SPACE.sub(" ", line) for line in reviews]
    return reviews

df_amazon.loc[:,'review'] = df_amazon['review_headline'] + ' ' + df_amazon['review_body']
df_amazon.loc[:,'review'] = preprocess_reviews(df_amazon.review)
df_amazon.loc[:,'star_rating'] = df_amazon.star_rating.astype(int)
```

In [6]:
```
df_train, df_val = train_test_split(df_amazon,
                                    test_size=0.2,
                                    random_state=2020,
                                    stratify=df_amazon.star_rating)
```

Sentiment Analysis

Sample data & distribution of ratings

In [42]:
```
df_amazon.sample(10, random_state=2020)
```

Out[42]:

	star_rating	review_headline	review_body	review
1792302	5	Five Stars	Thanks, this is a true value. Stylus is short...	five stars thanks this is a true value stylus...
7967590	5	grandaughter loves it	This is the second phone case I bought for my ...	grandaughter loves it this is the second phone...
6647536	1	screen and body	Not the greatest screen protector but for .99 ...	screen and body not the greatest screen protec...
6414044	4	its ok	Im very pleased with the product it does it jo...	its ok im very pleased with the product it doe...
7701488	5	A++	+ + + + + LOVE THIS PRODUCT IT'S AMAZING + + +...	a++ + + + + + love this product its amazing + ...
2437007	5	Love it!	I love it, works perfect. You can see on this ...	love it i love it works perfect you can see on...
6735977	4	Polka dot gel case	It's very cute. Easy to get off and on. I like...	polka dot gel case its very cute easy to get o...
5885602	5	My items I purchased	I loved this case and it fits real well on my ...	my items i purchased i loved this case and it ...
1546974	5	Love the colors	Love the colors! Offers a secure connection w...	love the colors love the colors offers a secu...
3278886	2	Two Stars	It did not fit properly in my car.	two stars it did not fit properly in my car

	Count	Percent
1	13906	14%
2	6474	6%
3	8967	9%
4	16962	17%
5	53691	54%

Sentiment Analysis

- Recall ULMFiT:
  - ✓ Preprocess text data
  - ☐ Create a language model to encode your text
  - ☐ Create a model (e.g. classifier) which uses the encoder you made

Modeling – Create a language model (2 – DataBunch)

In [5]:
```
from fastai.text import *
```

Create from dataframe

In [7]:
```
data = (TextList.from_df(df_amazon, cols='review')
        .split_by_rand_pct(0.2, seed=2020)
        .label_for_lm()
        .databunch(bs=32))
data.show_batch()
```

Use the column 'reviews' as text data

Label for a language model

Create a DataBunch with batch size 32

idx	text
0	feel and overall quality of the whole case everything is where it should be and i never forget my credit card or i d xxunk leather only feels better after a few days of being broken in and the slim that accompanies it is great because it does nt obstruct side swipe at all and allows for the nexus design to be fully utilized clasp is totally unnecessary especially since
1	iphone holder specially that it fits my otterbox case < br > i use it every day on my motorcycle xxbos new battery this was exactly what we needed but could nt get at our local cell phone store the old battery would not hold a charge for even a day we were excited that the new battery came fully charged and ready to be put in the phone will

Modeling – Create a language model

Create LM
Pretrained AWD LSTM

In [27]:
```
learn = language_model_learner(data, AWD_LSTM, drop_mult=0.3)
learn.model.cuda()
learn.fit_one_cycle(10)
learn.save_encoder('enc-stage-1')
```

Train on GPU
Fit one cycle over 10 epochs
Save encoder weights

epoch	train_loss	valid_loss	accuracy	time
0	4.566915	4.403250	0.214291	06:07
1	4.252519	4.160347	0.236389	06:09
2	4.200518	4.098262	0.242405	06:09
3	4.146877	4.075602	0.244574	06:10
4	4.078557	4.061365	0.246053	06:11
5	4.039335	4.050590	0.247892	06:09
6	4.028378	4.042146	0.248613	06:09
7	3.969337	4.035912	0.249497	06:09
8	3.936812	4.033516	0.250018	06:09
9	3.927001	4.033215	0.250072	06:09

Expect accuracies like this for LM's

Modeling – Create a language model

In [31]:
```
learn = language_model_learner(data, AWD_LSTM, drop_mult=0.3)
learn.load_encoder('enc-stage-2')
print(learn.predict('The product was', 5))
print(learn.predict('My shipment arrived', 5))
```

```
The product was delivered late but i noticed
```

input output

```
My shipment arrived on time and with a
```

input output

Sentiment Analysis

- Recall ULMFiT:
  - ✓ Preprocess text data
  - ✓ Create a language model to encode your text
  - ☐ Create a model (e.g. classifier) which uses the encoder you made

Modeling – Create a classifier (2 – DataBunch)

Create from dataframe

Use the vocabulary of our LM data

Label for a classifier (from a df)

```
In [8]: val_data = TextList.from_df(df_val, cols='review', vocab=data.vocab)

        data_clas = (TextList.from_df(df_train, cols='review', vocab=data.vocab)
                .split_by_rand_pct(0.2, seed=2020)
                .label_from_df(cols='star_rating')
                .add_test(val_data)
                .databunch(bs=32))

        data_clas.show_batch()
```

Token for start of a sentence

Token for unknown word

text	target
xxbos a comfortable professional looking headset with a high quality microphone & speaker exceptional value but not quite perfect for those wanting a short review < br > < br > this review has turned out to be the longest i have ever written so i thought a short version might be in order for those with limited attention spans … i mean limited time ☐ simply put the speaker	5
xxbos current monitor results for nexus 7 tab galaxy nexus phone samsung galaxy 77 tablet and ipad mini xxunk 42 amp is a two usb ports wall charger with retractable prongs to test this external battery i used a current monitor in addition to the current monitor i also used a 10 minute charging test to provide a less abstract measurements of this charger these tests allow me to compare	5

Modeling – Create a classifier (3 – Initial Fit)

Classifier Learner

```
In [32]: learn_classifier = text_classifier_learner(data_clas, AWD_LSTM, drop_mult=0.5)
learn_classifier.load_encoder('enc-stage-2')
learn_classifier.freeze()
learn_classifier.fit_one_cycle(10)
learn_classifier.show_results()
```

Use encoder from LM

epoch	train_loss	valid_loss	accuracy	time
0	0.847336	0.695352	0.725250	04:07
1	0.815760	0.653389	0.747312	8:30:19
2	0.801872	0.648235	0.752437	04:05
3	0.803606	0.642513	0.748062	04:04
4	0.754694	0.626177	0.754125	04:13
5	0.715627	0.628537	0.753313	03:59
6	0.699694	0.621820	0.759562	03:57
7	0.744746	0.619439	0.757563	03:58
8	0.722054	0.621591	0.758563	04:02
9	0.747256	0.615466	0.759625	04:01

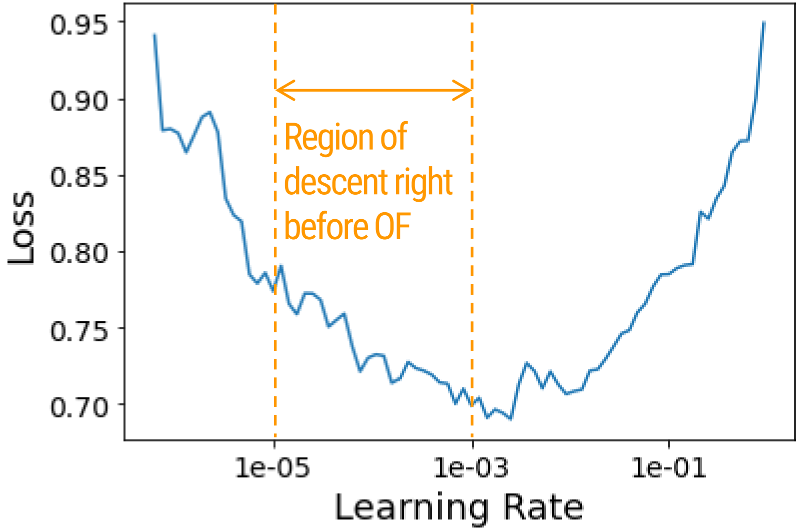
Sentiment Analysis

Modeling – Create a classifier

	text	target	prediction
	xxbos a nice phone but some drawbacks i got a chance to play around with an htc one m9 for the last couple of days i am a complete phone nerd and a recognized xxunk on xda developers where all kinds of android phone modifications are developed and discussed if that means anything to you so when i get a chance to play around with a new phone i jump	4	3
	xxbos a great radar detector update feb 23 2014 < br > xxrep 19 _ i thought it only fair to update this review escort did rma my detector and replaced the gps chip in it i was without for xxunk 3 weeks but now i m using it daily since mid january and i can say it has performed excellently i love the auto xxunk of false alarms and	4	1
	xxbos great unit for great price so far i have had this unit in for 1 day in my truck and i can tell you that the sound improvement over the stock system is nothing short of amazing for starters i have a 2006 gmc sierra it comes with the gm head unit onstar integrated and the bose amplifier for those in a similar situation as i you will have	5	5
	xxbos makes iphone bulky but definitely is well protected i bought a factory unlocked iphone 4 and lets just say it was not cheap $ 1k coupled with the fact that i have a 15 month old son and a little clumsy well i definitely wanted to have my iphone super duper protected i ordered from amazon and had the free shipping amazon is great i got it relatively fast	5	4
	xxbos ampen new hybrid stylus an updated product that works well using a stylus with various devices can be highly subjective but when you find one that works well for you its hard to do without it this one is a good example of taking a first rate product and making subtle improvements to make it truly excellent the xxunk ampen new hybrid stylus silver is an example of product	5	5

Modeling – Create a classifier (4 – Unfreeze & run LR Finder)

In [132]:
```
learn_classifier.freeze_to(-2)
learn_classifier.lr_find()
learn_classifier.recorder.plot()
```



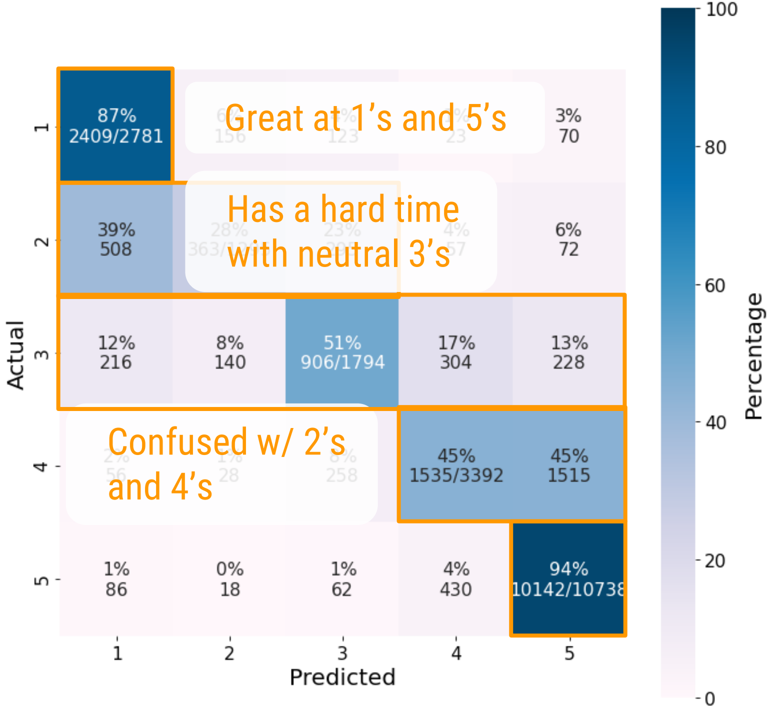
(5 – Tuning Fit)

In [133]:
```
learn_classifier.fit_one_cycle(5, slice(1e-5, 1e-3))
```

epoch	train_loss	valid_loss	accuracy	time
0	0.693067	0.600351	0.764000	04:36
1	0.683088	0.577081	0.769437	04:33
2	0.687134	0.566117	0.773750	04:33
3	0.608367	0.567810	0.774312	04:32
4	0.627051	0.567901	0.772563	04:31

Sentiment Analysis

Modeling – Interpret results (6)




```
In [18]: preds, target = learn_classifier.get_preds(DatasetType.Test, ordered=True)
         preds = np.argmax(preds, axis=1)
         preds += 1
```

```
In [39]: accuracy_score(df_val.star_rating, preds)
Out[39]: 0.76775
```

Sentiment Analysis

Modeling – Interpret results (note: add 1 to the prediction)

```
In [19]: learn_classifier.predict("Five stars")

Out[19]: (Category tensor(4),
          tensor(4),
          tensor([0.0048, 0.0185, 0.0280, 0.0257, 0.9230]))

In [83]: learn_classifier.predict("Good product, however the shipping was delayed")

Out[83]: (Category tensor(2),
          tensor(2),
          tensor([0.1092, 0.2453, 0.3923, 0.1468, 0.1063]))

In [84]: learn_classifier.predict("Would not recommend")

Out[84]: (Category tensor(0),
          tensor(0),
          tensor([0.6133, 0.3287, 0.0368, 0.0127, 0.0085]))

In [88]: learn_classifier.predict("To the left to the left, everything you own in the box to the left")

Out[88]: (Category tensor(0),
          tensor(0),
          tensor([0.2724, 0.2589, 0.2586, 0.0678, 0.1423]))
```

Image Classification

- Create dataset
  - Downloaded 322 images for each category from Google images (scraping details are in notebook)

Here are the categories we'll be modeling in this notebook:

- Men's shoes (shoes_men)
- Women's shoes (shoes_women)
- Socks (socks)
- Leggings (leggings)

```
.
├── data
│   ├── footwear
│       ├── shoes_men
│       ├── shoes_women
│       ├── socks
│       ├── leggings
```

Image Classification

Create DataBunch (1 & 2)

```
In [4]: np.random.seed(42)
        data = ImageDataBunch.from_folder(path,
                                          train=".",
                                          valid_pct=0.2,
                                          ds_tfms=get_transforms(),
                                          bs=bs,
                                          size=224
                                          ).normalize(imagenet_stats)
```

Typical image transforms (rotation, flipping, etc)

Normalize pixel values

```
In [5]: data.classes
```

```
Out[5]: ['leggings', 'shoes_men', 'shoes_women', 'socks']
```

We use the function `.show_batch()` to show sample images from our DataBunch.

```
In [6]: data.show_batch(rows=4, figsize=(8,8))
```

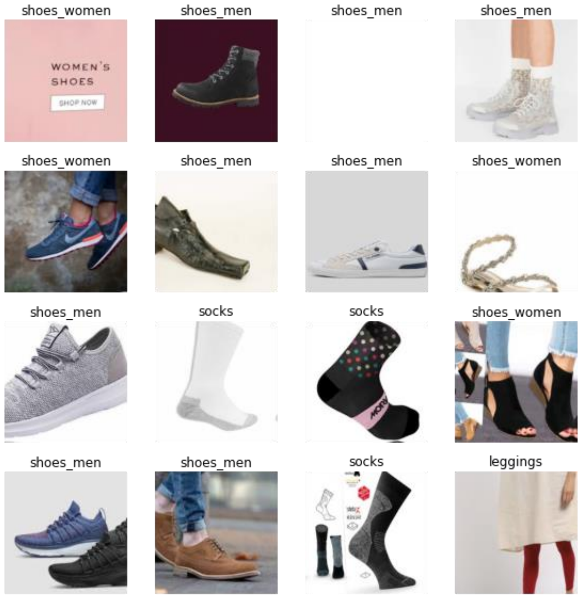


Image Classification

Perform initial fit (3)

```
In [7]: learn = cnn_learner(data, models.resnet50, metrics=error_rate)
        learn.model.cuda()
```

Pretrained ResNet50 model

```
In [8]: learn.fit_one_cycle(10)
```

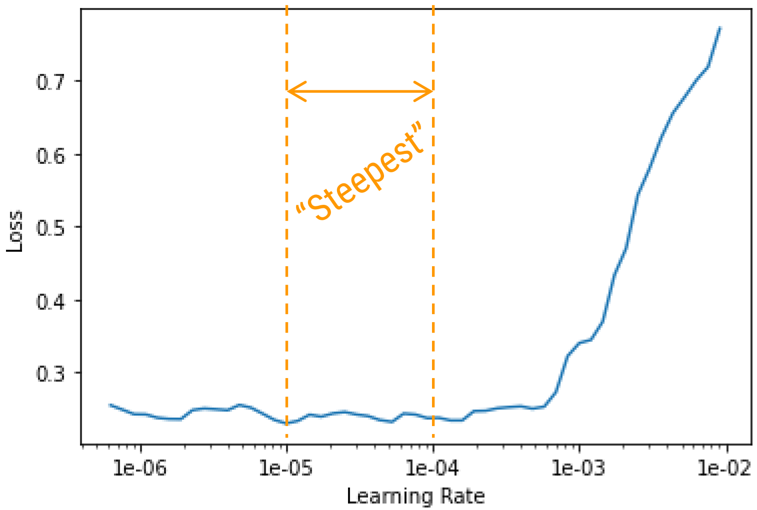
epoch	train_loss	valid_loss	error_rate	time
0	1.303267	0.797555	0.272374	01:22
1	1.019082	0.718676	0.245136	01:18
2	0.878435	0.639344	0.225681	01:19
3	0.726180	0.551344	0.210117	01:18
4	0.601773	0.548616	0.217899	01:17
5	0.511514	0.505609	0.202335	01:17
6	0.422099	0.446399	0.178988	01:17
7	0.365781	0.461301	0.217899	01:17
8	0.326806	0.449308	0.194553	01:20
9	0.289613	0.458108	0.186770	01:19

Image Classification

Perform tuning fit (4 & 5)

In [10]:
```
learn.unfreeze()
```

In [11]:
```
learn.lr_find()
learn.recorder.plot()
```



In [12]:
```
learn.fit_one_cycle(5, max_lr=slice(1e-5,1e-4))
```

epoch	train_loss	valid_loss	error_rate	time
0	0.247169	0.442216	0.190661	01:23
1	0.234456	0.463489	0.210117	01:23
2	0.213014	0.483345	0.198444	01:21
3	0.189413	0.470113	0.194553	01:22
4	0.171666	0.463862	0.194553	01:22

Save models

In [17]:

```
start = 22
learn.fit_one_cycle(30,
                    start_epoch = start,
                    callbacks=[SaveModelCallback(learn,
                                                 every='epoch',
                                                 monitor='error_rate',
                                                 name='resnet50_30te_cutout_mixup')])
```

30 total epochs

Save model weights every epoch

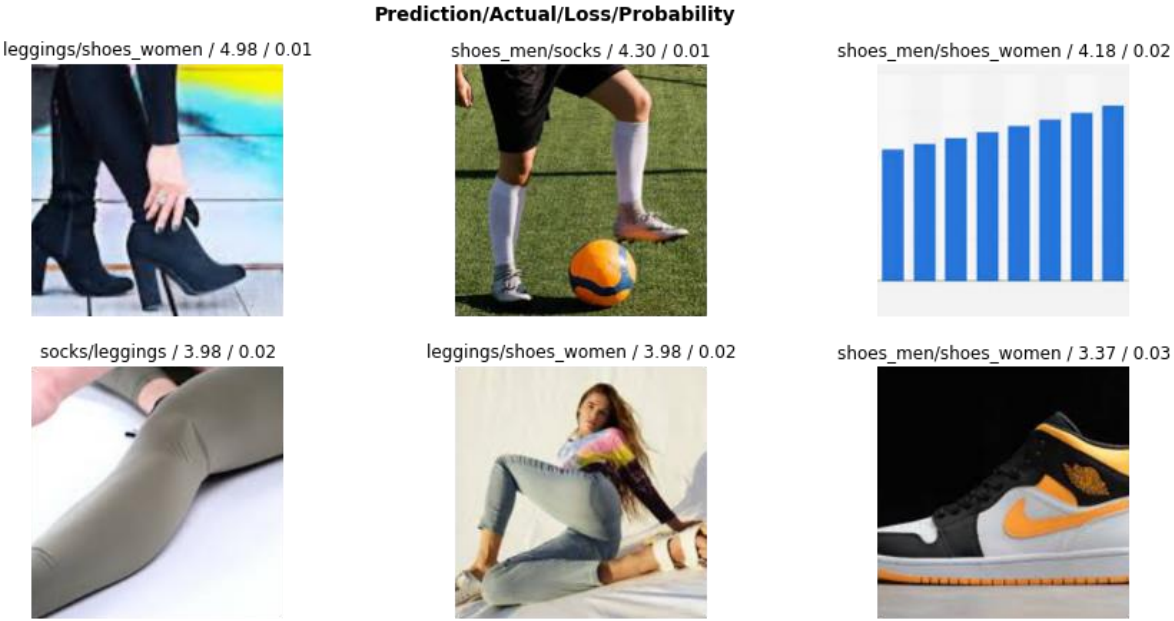
```
Loaded resnet50_30te_cutout_mixup_21
```

epoch	train_loss	valid_loss	error_rate	time
22	1.648879	0.829137	0.217241	29:56
23	1.599334	0.824483	0.215912	31:04
24	1.604851	0.821494	0.214489	32:15
25	1.593274	0.815072	0.214157	31:56
26	1.546791	0.815933	0.214204	32:49
27	1.537701	0.812505	0.214109	34:24
28	1.574610	0.814327	0.213066	34:30
29	1.568347	0.808587	0.212781	36:12

Image Classification

Interpret Results (6)

Top Losses



Most Confused

```
Most confused categories (Actual, Predicted, Count):
[('shoes_men', 'shoes_women', 26), ('shoes_women', 'shoes_men', 11)]
```

Image Classification

Interpret Results

Confusion Matrix

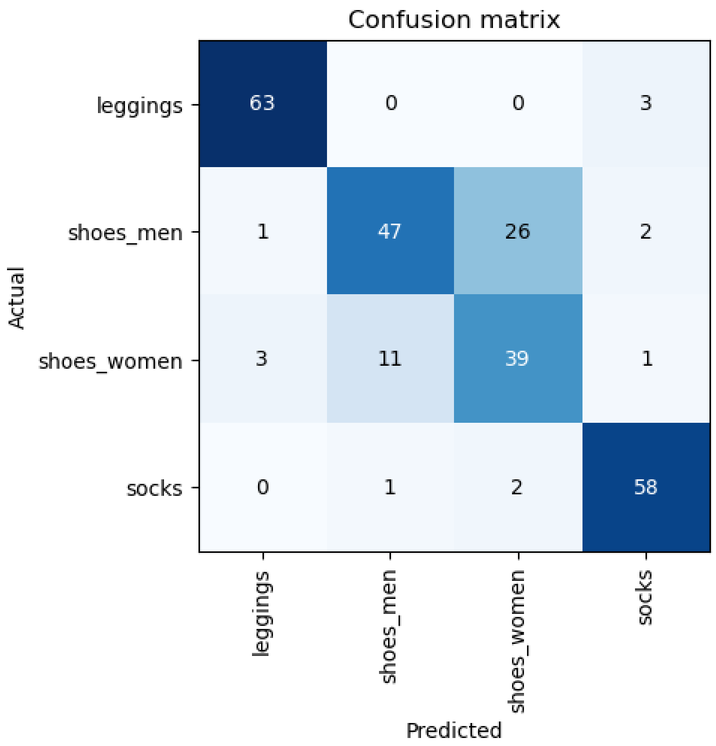
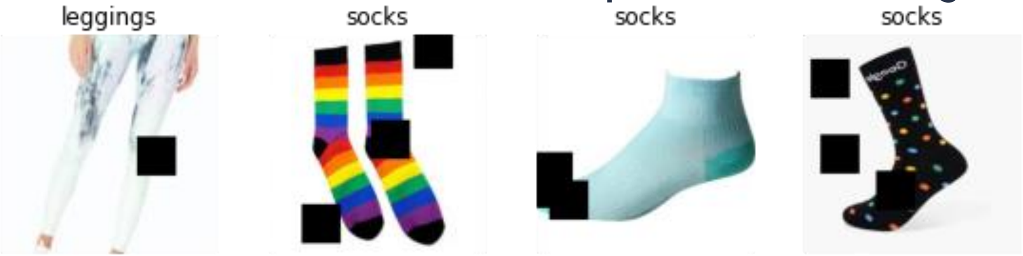


Image Classification

- Some regularization techniques
  - **Cutout** – remove random squares from images



  - **Mixup** – overlay two semi-transparent images

  - **Label smoothing** – use decimal labels (0.9, 0.1)

Image Classification

- Some regularization techniques - implementation
  - **Cutout** – remove random squares from images

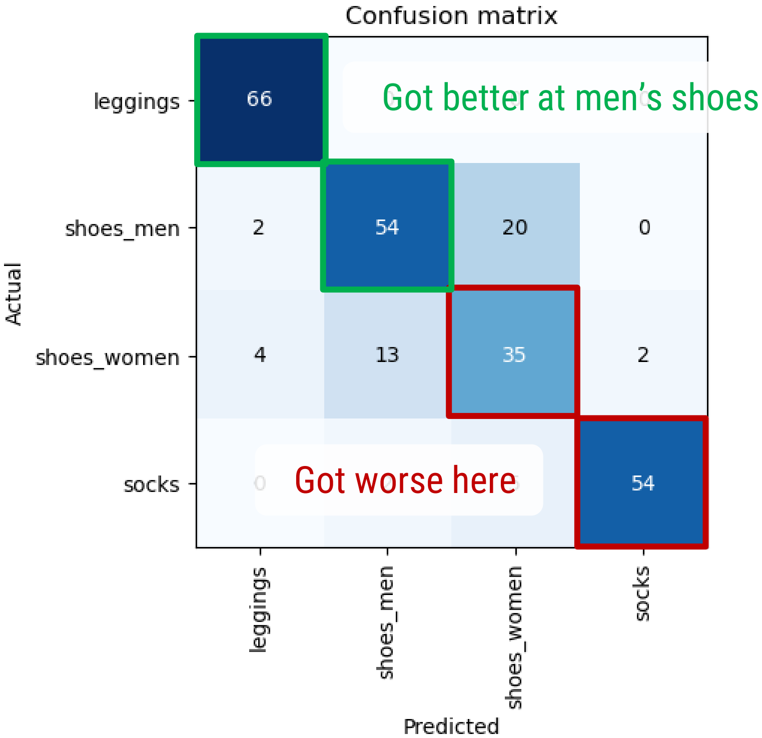
```
tfms = get_transforms(xtra_tfms = cutout(n_holes=(1,3)))
```

  - **Mixup** – overlay two semi-transparent images
  - **Label smoothing** – use decimal labels (0.9, 0.1)

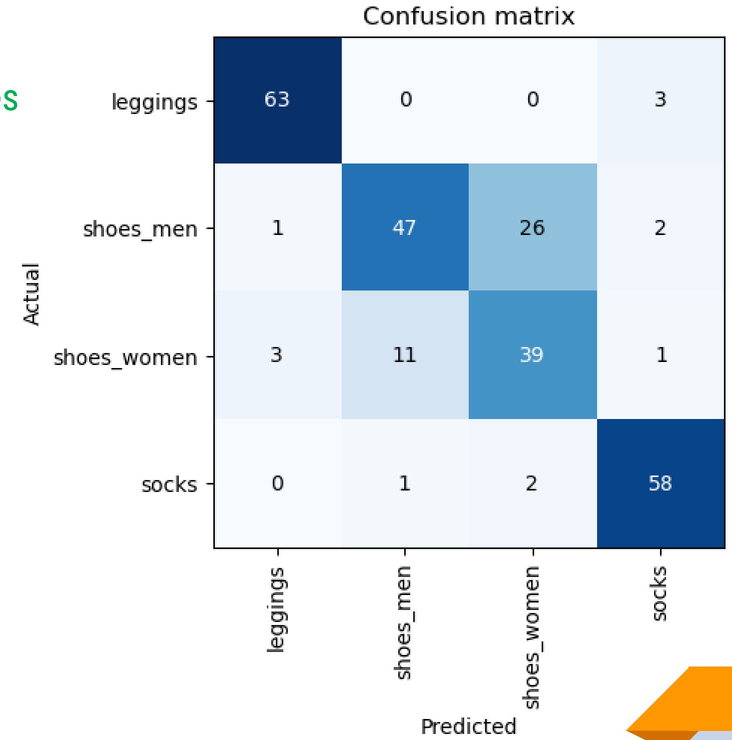
```
In [15]: learn = cnn_learner(data,
                             models.resnet50,
                             metrics=error_rate,
                             loss_func=LabelSmoothingCrossEntropy()).mixup()
         learn.model.cuda()
         learn.fit_one_cycle(10)
         learn.save('ls_mixup_stage-1')
```

Image Classification

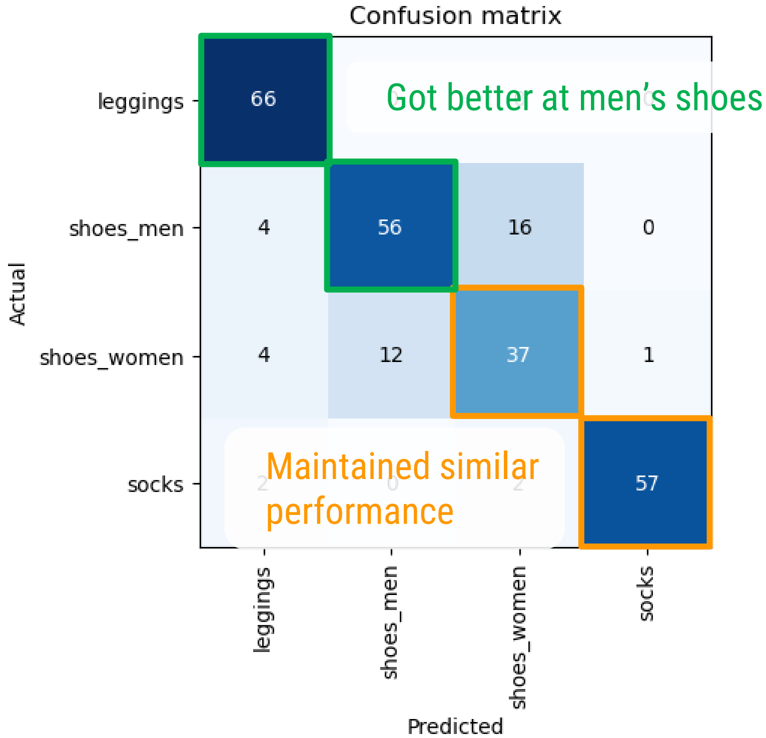
Results of **Cutout**



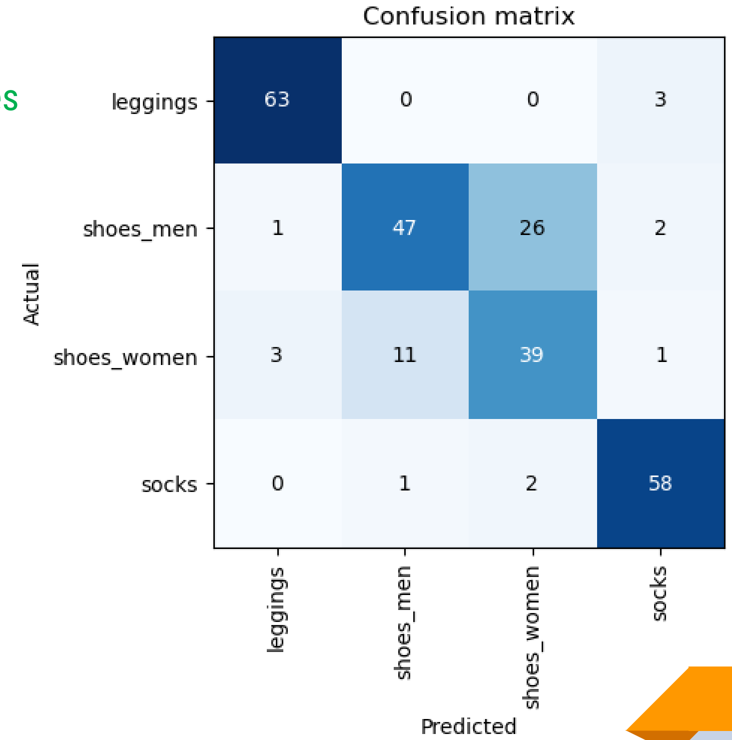
Original model

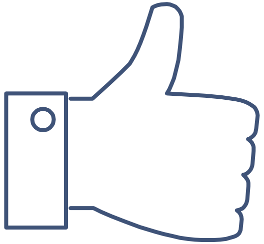


Results of **Mixup & LS**



Original model





Thank you!

References

- FastAI Docs, Forum, and Courses
- Jeremy Jordan – Setting the learning rate of your neural network
- Subhajit Das – The 1Cycle Policy: an experiment that…
- Sylvain Gugger – The 1Cycle Policy
- Gilbert Tanner – FastAI Sentiment Analysis

CREDITS

Special thanks to:

- Presentation template by SlidesCarnival
- Photographs by Startup Stock Photos