



# Deep Learning for Coders



Fast.ai



# Fast Ai Library

---

- Jeremy Howard and Rachel Thomas
- Built on Pytorch
- Built to run on GPU
- Literally fast

# Image Classification - Cats and Dogs

---

```
arch=resnet34
```

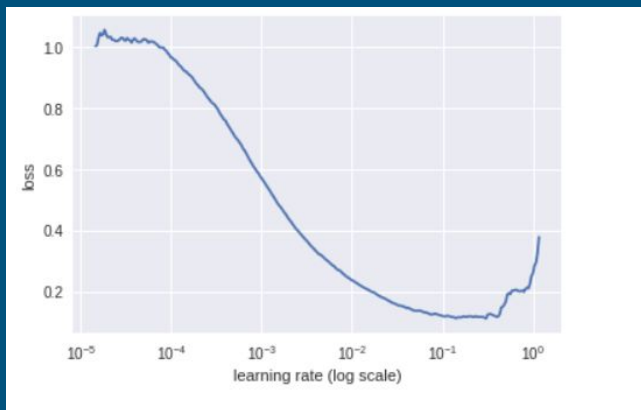
```
data = ImageClassifierData.from_paths(PATH,  
tfms=tfms_from_model(arch, sz))
```

```
learn = ConvLearner.pretrained(arch, data, precompute=True)
```

```
learn.fit(0.01, 2)
```

# Learning Rate (LR)

- Gradient descent for optimizing function
  - Stochastic gradient descent
  - Batch gradient descent
  - Mini - batch gradient descent
- `learn.lr_find()`
- Loss Vs. LR

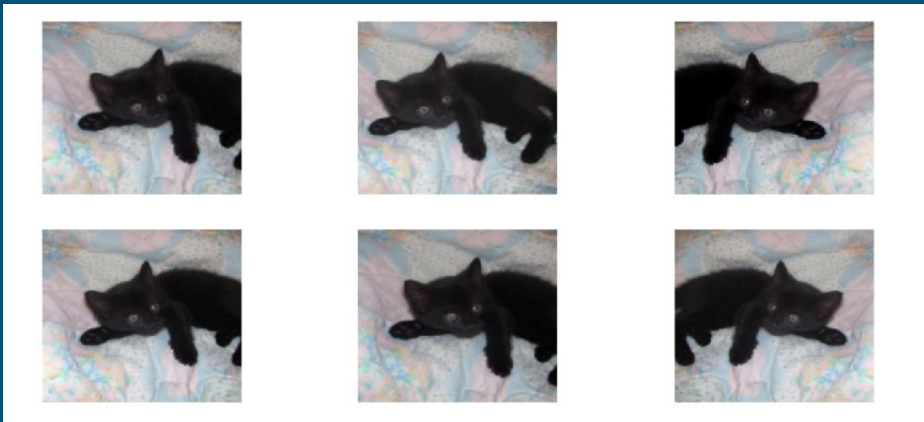


# Improving Model

---

- Data Augmentation

```
tfms = tfms_from_model(resnet34, sz,  
aug_tfms=transforms_side_on, max_zoom=1.1)
```



# Improving Model

---

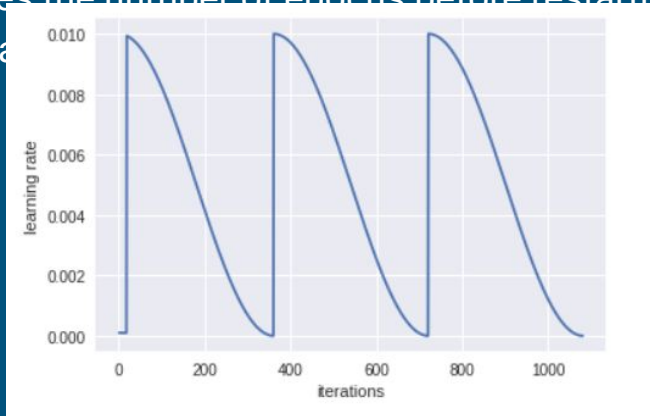
- Data Augmentation

- `Precompute = False`

- SGDR (SGD with Restarts)

- `Cycle_len` - specifies the number of epochs before restarting

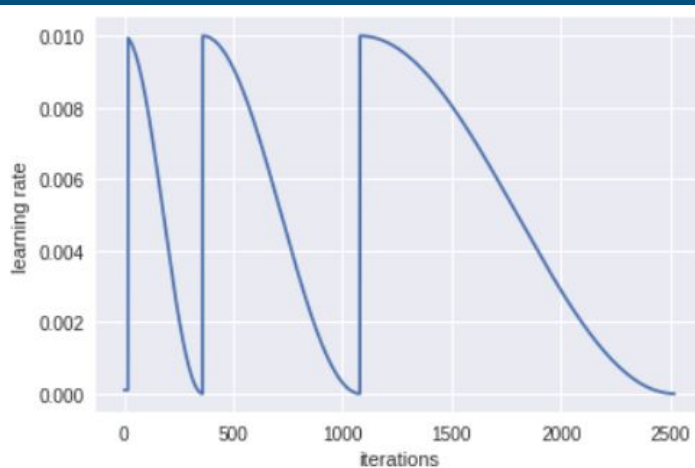
- A version of learning rate



# Improving Model

- Learning rate Annealing

- `learn.unfreeze()`
- `lr=np.array([1e-4,1e-3,1e-2])`
- `learn.fit(lr, 3, cycle_len=1, cycle_mult=2)`



# Improving Model

---

- Test Time Augmentation

```
log_preds, y = learn.TTA()
```

```
probs = np.mean(np.exp(log_preds), 0)
```

```
accuracy_np(probs, y)
```