

Fast AI lesson 6 : remaining part of 05_pets_breeds

06 July 2021 13:23

What really transfer learning does?

The transfer learning enables us to learn from the weights which are previously trained on some big dataset we throw away some last layers and then we train that on our data

Fine tuning the data

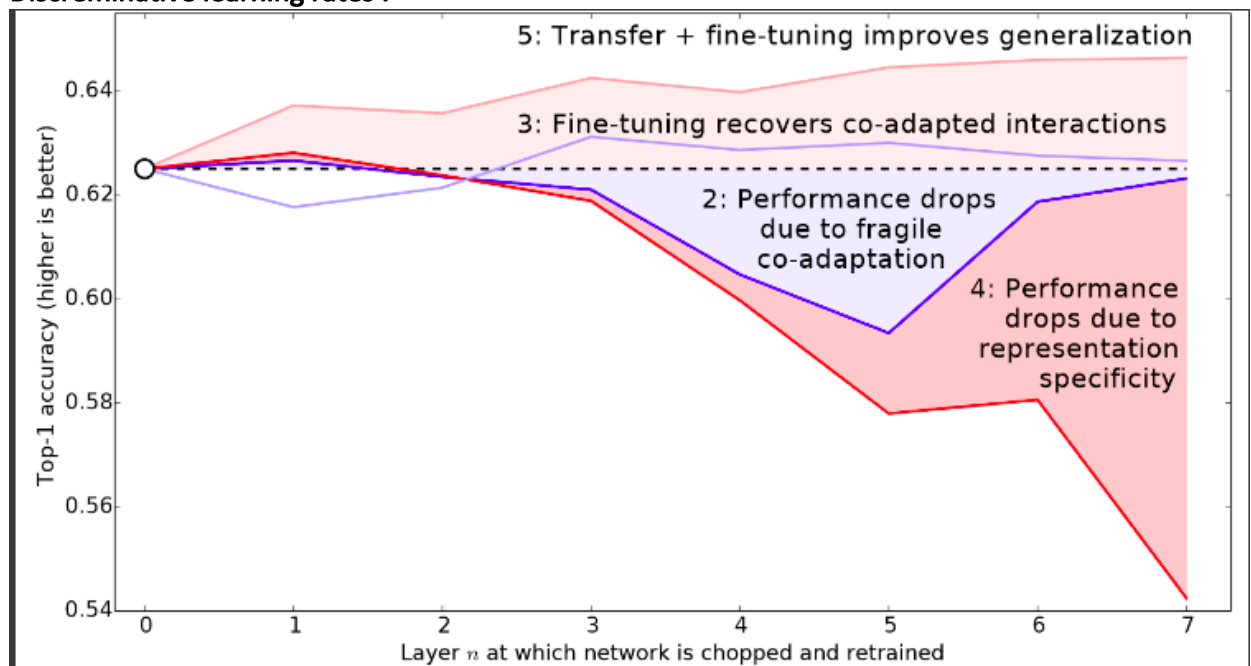
`Learner.fine_tune()`

Cnn_learner defaultly freezes the layers

In `learner.fine_tune` does the fine tuning with the half of learning rate for one epoch and the after that `fit_one_cycle` function gets called and the fitting on the afterwards epochs get's taken place

Small learning rate in beginning layers and the bigger learning rates in the later layers

Discreminative learning rates :



For working with the discreminative learning rate rather than passing the fixed learning rate we can pass the slice which will give the range of learning rates

For building with the discreminative learning rate we can do is passing the `lr_max` parameter while fitting the learner for `fit_one_cycle()`

`Learn.fit_one_cycle(12, lr_max = slice(1e-6, 1e-4))`

Where the `1e-6` will be the learning rate of the first layer and `1e-4` will be of the last one layer and the layers in between those two will have learning in equal multiplication they are responsible for equally spaced learning rate for layers

Fitonecycle is bit different than the fit starts at the low learning rate and increases the gradually for the 1st 3 batches the highest it can go for the range we mentioned then it again gradually decreases with the each epoch

Plotting the loss through hlearner
`Learn.recorder.plot_loss()`

How can we make it better we can get the more less error we can try deeper architecture

Deeper architecture ;

Putting the more pairs of the activation functions or the non linearity the number of the unit layers which have it's getting showed on the layer variants

In practice more of the times where we want to pick up from the selected models

For the bigger models since they have the more layers they are able to figure out more fine patterns inside it

If we have the more layers we might get the out of memory error from the GPU if that happens then we've to rerun all the notebook again

`To_fp_16()` does the 1/2 precision floating point and use the less memory on any nvidia 2020 or later as they give more efficient results in less amount of time

Summary of the chapter :

- Here in this chapter we have discussed about the crossentropy loss,
- Training of model
- Fine tuning the models
- What are dataloaders
- What are the data blocks
- How to get path
- How to use dataloaders to load the data efficiently
- Softmax and sigmoid functions (activations)
- Relu and leaky relu
-