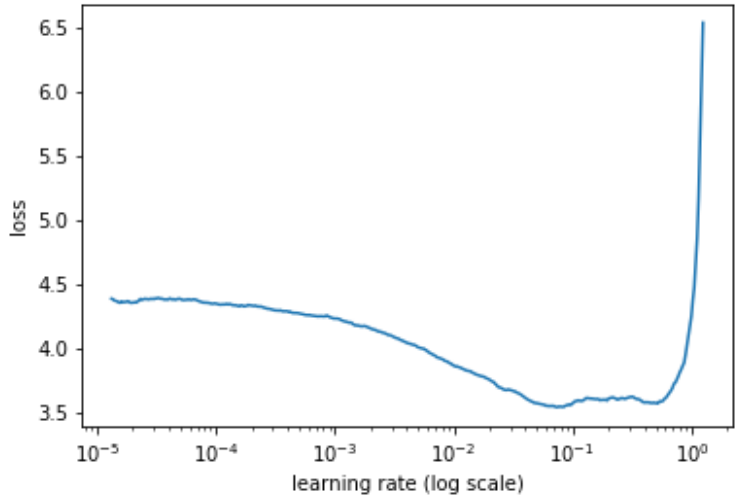
Notes from the course “Deep Learning for Coders” (2018)

Lesson 2

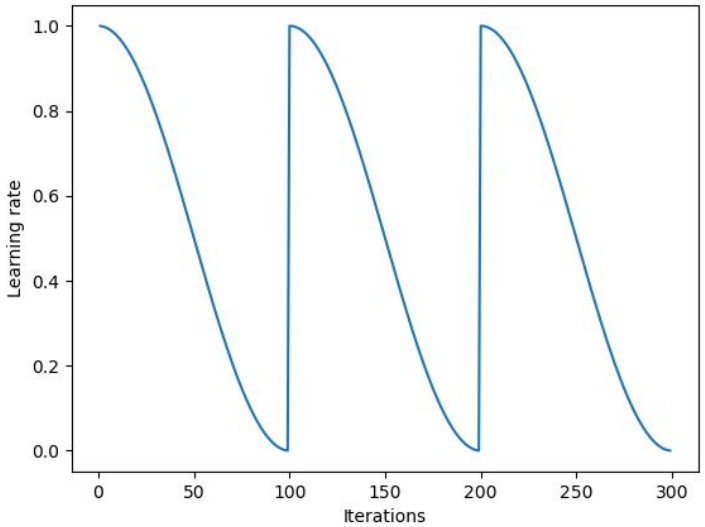
Automatic learning rate finder

One automatic way of finding a good learning rate is to start with a low learning rate and then increase it after each batch. After a while the loss will start to increase. Pick the learning rate where the loss decreases the most. In the example below, 1e-2 would be a good learning rate.

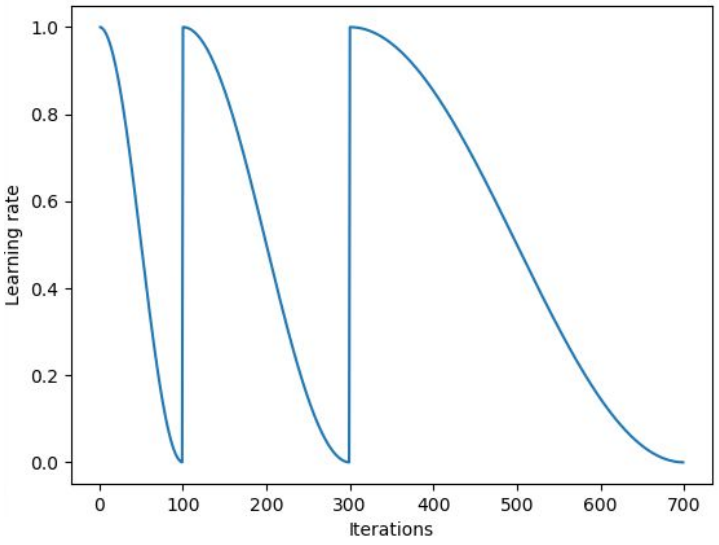


Cosine learning rate annealing with restarts

Cosine learning rate annealing changes the learning rate according to a cosine function with restarts. The rationale behind the restart is that we want minimums that generalize well. The restart helps us find these minimum by escaping “weak” minimums.



In the previous example the cycle length are constant. However, sometimes we want to increase the number iterations per cycle as function of iterations, as in the example below. This is especially true if the training loss is greater than the validation loss.



Differential learning rates

Sometimes we want different learning rates for different layers. The learning rate should most likely be lower for the layers in the beginning of the network.

For example: [1e-5, 1e-4, 1e-3], where 1e-5 is for the earliest layers, 1e-4 is for the middle layers and 1e-3 is for the fully connected layers. As we can see in the example above, the learning rate decreases with a factor of 10 for each section.

As a rule of thumb, they should decrease with a factor of 3-10x at each section, depending upon how similar they are to the original transfer model's data.

For example, assume that we have a model that is trained on ImageNet. If our task is to classify cats/dogs then we probably want to decrease it less, i.e. 3 times, since it is fairly similar to ImageNet dataset. However if we work with medical or satellite images we probably want to decrease it more, i.e. 10 times.

Transfer learning

When doing transfer learning, it is a good idea to freeze the feature extraction layers and find "okay" fully connected layers. Then we can unfreeze all layers and train them with differential learning rates as described above.

For the transfer model, ResNet34 trained on ImageNet is a good start. After that, one might want to try ResNext50.

Data augmentation

Try choosing data augmentation (i.e. flip vertically/horizontally etc) that makes "sense", flipping upside down does not make sense for cats/dogs but it does for satellite images. Another trick is to train on smaller images and switch to larger ones later, preventing overfitting. However, this assumes that we have used a model which does global pooling in the last layer.

Easy steps to train a good image classifier

1. Find the learning rate with the automatic learning rate finder.
2. Train the last layer with data augmentation for 2-3 epochs with a cycle length of 1.
3. Unfreeze all layers.
4. Set earlier layers to lower learning rate than the last layer.
5. Train the full network with a cycle length of 2 until we reach overfitting.