

1 Summary

- I decided to explore the cats-dogs-pandas data set:
<https://www.kaggle.com/ashishsaxena2209/animal-image-datasetdog-cat-and-panda>
- The full notebook where I loaded the data, built an alexnet model and predicted a sample can be explored here:
<https://colab.research.google.com/github/simonwardjones/d2l-study-group/blob/master/exercises/7-sw-j.ipynb>

2 Steps in notebook

- 1 Connect to kaggle api and download data
 - (a) Mount Google drive
 - (b) Install kaggle cli then download and unzip data
- 2 Explore the data and prepare dataset/dataloader
 - (a) Use the torchvision datasets.ImageFolder helper to define dataset
 - (b) Use torchvision Resize transformer and visually inspect a few samples
- 3 Train test split data
 - (a) Use sklearn train_test_split to stratify (same proportion of animals in train and test)
 - (b) Define batch_size, learning rate and num epochs
 - (c) Create data loaders for train/test using SubsetRandomSampler
 - (d) use Resize and ToTensor transforms
 - (e) Note a batch has size [64, 3, 224, 224] 64 samples, 3 input channels, height and width 224
- 4 Define AlexNet structure
 - (a) I used the same architecture as alex net
 - (b) Throughout the net channels increase as height and width decrease using sequential convolutions (of decreasing kernel sizes) and max pooling (after 1st, 2nd and 5th convolutions).
 - (c) The Relu activation is used throughout after each convolution and before max-pooling
 - (d) The final three layers are fully connected
- 4 Train model
 - (a) Train model using animator to visualise loss and test accuracy
 - (b) If already trained load from gdrive
 - (c) Save the model using torch. Save and cp to drive
- 4 Predict examples
 - (a) Predict examples and show images

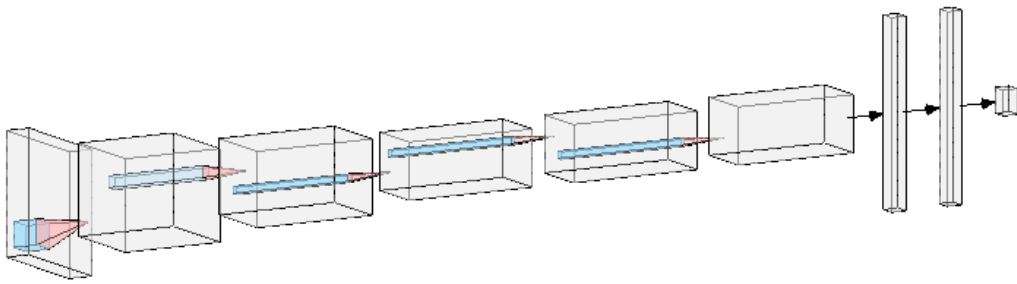


Figure 1: AlexNet architecture