Data Augmentation

"Deep learning is only relevant when you have a huge amount of data". While not entirely incorrect, this is somewhat misleading. Certainly, deep learning requires the ability to learn features automatically from the data, which is generally only possible when lots of training data is available --especially for problems where the input samples are very high-dimensional, like images. However, convolutional neural networks --a pillar algorithm of deep learning-- are by design one of the best models available for most "perceptual" problems (such as image classification), even with very little data to learn from. Training a convnet from scratch on a small image dataset will still yield reasonable results, without the need for any custom feature engineering. Convnets are just plain good. They are the right tool for the job.

We have to reduce the amount of irrelevant features in the dataset. You can just flip the images in the existing dataset horizontally such that they face the other side! Now, on training the neural network on this new dataset, you get the performance that you intended to get. By performing augmentation, can prevent your neural network from learning irrelevant patterns, essentially boosting overall performance.

Where do we augment data in our ML pipeline?

- The first option is known as offline augmentation. This method is preferred for relatively smaller datasets, as you would end up increasing the size of the dataset by a factor equal to the number of transformations you perform (For example, by flipping all my images, I would increase the size of my dataset by a factor of 2).
- The second option is known as online augmentation, or augmentation on the fly. This method is preferred for larger datasets, as you can't afford the explosive increase in size. Instead, you would perform transformations on the mini-batches that you would feed to your model. Some machine learning frameworks have support for online augmentation, which can be accelerated on the GPU.

There are a variety of data augmentation methods. The specific techniques used for augmenting data depend upon the nature of data with which a user is working. Note that data augmentation is typically implemented during preprocessing on the training dataset.

Advanced models for data augmentation are:

- Adversarial training/Adversarial machine learning: It generates adversarial examples which disrupt a machine learning model and injects them into a dataset to train.
- Generative adversarial networks (GANs): GAN algorithms can learn patterns from input datasets and automatically create new examples which resemble training data.
- Neural style transfer: Neural style transfer models can blend content image and style image and separate style from content.
- Reinforcement learning: Reinforcement learning models train software agents to attain their goals and make decisions in a virtual environment.

Classic image processing activities for data augmentation are:

- padding
- random rotating
- re-scaling,
- vertical and horizontal flipping
- translation (image is moved along X, Y direction)
- cropping
- zooming
- darkening & brightening/color modification
- grayscaling
- changing contrast
- adding noise
- random erasing

Benefits of data augmentation include:

- Improving model prediction accuracy
    - adding more training data into the models
    - preventing data scarcity for better models
    - reducing data overfitting (i.e. an error in statistics, it means a function corresponds too

closely to a limited set of data points) and creating variability in data

- increasing generalization ability of the models
- helping resolve class imbalance issues in classification

- Reducing costs of collecting and labeling data
- Enables rare event prediction
- Prevents data privacy problems