Department of Computing

Imperial College of Science, Technology and Medicine

Data Augmentation

Bare-bones summary of the lectures

Author: Anton Zhitomirsky

Contents

1 Data Augmentation

1.1 Input Augmentation

Input data augmentation is a technique used to increase the size and diversity of your training set. It does this by applying a series of random but realistic transformations to each data point during training

By increasing the size of your training data artificially, you're effectively adding more "experiences" for your model, which can improve generalization.

It can be done through imgaug or pytorch.transforms.

One of the main benefits of data augmentation is its role as a regularizer. It helps prevent overfitting by ensuring that the model encounters a variety of different, yet plausible, examples during training.

1.2 Transformations

- Random
  - flipping - can simulate the natural orientations of objects or scenes in images
  - scaling - It helps the model generalize across different sizes of the same object or feature.
  - rotations - add another layer of complexity by altering the angle of the data points. This is especially useful in tasks like object recognition, where orientation can vary widely.
  - intensity/contrast variations - different lighting conditions and image qualities
  - cropping/padding - alter the focus and frame of the data points. This can be helpful for tasks where the subject can be off-center or partially visible
  - noise - serves as an effective way to improve the model's robustness. It mimics real-world scenarios where data may not always be clean or noise-free
  - affine transformations - translation, scaling, and shearing. These offer another way to introduce variability into your data, helping your model generalize better
  - perspective transformations - alter the viewpoint of the object or scene. This helps in applications like augmented reality, where the perspective can dramatically change the appearance of object

1.3 Anomaly detection

Pick out the unusualities out of one type of class.

1.3.1 Predicit Continuation

You would train something that is able to regress, to predict the continuation of the function. Then you train some sort of auto encoder that is able from a limited set of inputs to continue this function. If you continue this function and the prediction is too different from your external observation, you reflect that it is an outlier.

1.3.2 Measure distance in Latent Space

TODO

1.3.3 Reconstruct the input

If you have an auto-encoder, you need to take an input image and reconstruct it. After training, in theory, the network would have a hard time reconstructing something that it has never seen - so it will likely remove it from the image, which means that if you then take the output reconstruction and subtract from the external input which has the anomaly then the error will get highlighted.

1.3.4 Classify artificial, subtle variations - out of distribution detection

1.4 Approaches

1.4.1 Unsupervised

Use auto-encoder reconstruction error and use moving averages, dropout and set time window

1.4.2 Supervised

RNNs Learn form a set of yes/nos ina time series. RNNs can learn from a series of time steps and predict when the anomaly is about to occur.

1.4.3 Streaming and minibatches