Ideas for Exoplanet Classification

Kristyn Pantoja

1/15/2020

Neural Network Architecture

Data

The data is 4 time series for each observed object (to be classified as an exoplanet, or not).

Architecture

1. Import the data somehow, in some form.
2. LSTM
3. Pass it to an encoder, which learns a distribution over each of the two classes
4. Pass it to a decoder
5. loss is binary crossentropy + sum of KL divergences

Autoencoder

Why autoencoder? It can model non-linear relationships better than PCA (with one dimensional latent space)[1]

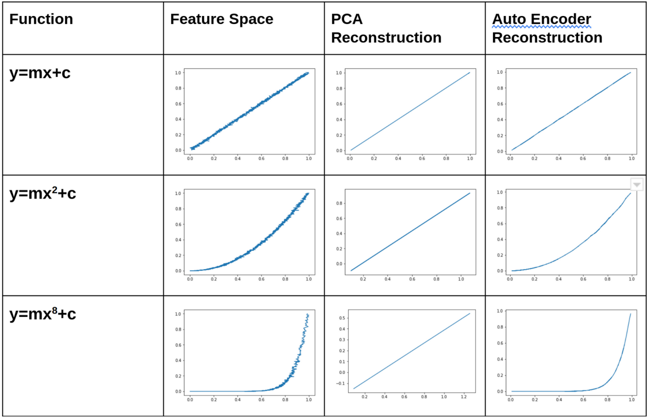

Figure 1: One-Dimensional Reconstruction PCA vs Autoencoder

[1] https://towardsdatascience.com/autoencoders-vs-pca-when-to-use-which-73de063f5d7

There are extensions of PCA to model non-linear relationships (Kernel PCA) which might be an interesting alternative, if autoencoders isn't the right move.

Autoencoders are also used for rare events classification[2], in a method similar to anomaly detection. Perhaps these methods would be harder to apply to exoplanet detection, though, since the negative labels would be mostly noise (hard to learn a distribution for)

Note: There is also a neat connection between PCA and autoencoders: "When the encoder and decoder are linear and L is the mean squared error, an undercomplete autoencoder learns to span the same subspace as PCA."[3]

[2] https://arxiv.org/abs/1809.10717
[3] https://arxiv.org/abs/1804.10253

Variational Autoencoder

Instead of just learning a compressed version of the data, we can learn the (variational) parameters of a probability distribution $q(z|x)$ that can be used to generate the data.

This will make the autoencoder more robust, as we can then train it on samples generated from the learned distribution.

A common distribution to learn is multivariate normal (dimension equal to that of the compressed data), and we learn the mean and variance parameters.

LSTMs

LSTM

Instead of using fully connected layers for the encoder and decoder, we can use LSTMs to also learn the temporal structure (if the data allows)

Bidirectional LSTM

Learn the sequence forward and backward (reversal)

Autoencoders in Clasification

We stack an autoencoder ontop of a classifier[4]

1. Train the autoencoder (its layers) on MSE loss - no need for labels here.
2. Use the learned latent representation (freeze) to compress the inputs of the classifier, which is a separately-trained SoftMax layer, to be trained separately, on crossentropy loss (using the labels)[5].
3. Then, if you want, retrain them altogether (called "fine-tuning").

[4] https://www.kaggle.com/shivamb/semi-supervised-classification-using-autoencoders

[5] https://stats.stackexchange.com/questions/318952/training-autoencoder-with-softmax-layer

Some Other Cool Stuff

Jigsaw Puzzles

Solving Jigsaw Puzzles[6]: "... we build a CNN that can be trained to solve Jigsaw puzzles as a pretext task, which requires no manual labeling, and then later repurposed to solve object classification and detection... By training the CFN to solve Jigsaw puzzles, we learn both a feature mapping of object parts as well as their correct spatial arrangement. Our experimental evaluations show that the learned features capture semantically relevant content."

MusicVAE

Blending Music[7]: "recurrent VAE models have difficulty modelingsequences with long-term structure. To addressthis issue, we propose the use of ahierarchicaldecoder, which first outputs embeddings for sub-sequences of the input and then uses these em-beddings to generate each subsequence indepen-dently."

[6] https://arxiv.org/abs/1603.09246
[7] https://arxiv.org/abs/1803.05428