Representation Learning Using Gaussian Processes

Arnav Arnav
Indiana University Bloomington
School of Informatics and Computing
aarnav@iu.edu

Abstract

Representation learning is the problem of finding a latent representation from the data that is compact and can still encode important information about the data. If such a representation can be obtained, it can then be used for subsequent tasks such as regression or classification. Obtaining such representations can be difficult and there has been much research in this field. Gaussian Processes have been used to model complex functions to represent the data in both classification and regression scenarios and have been known to scale well for large datasets. The focus of this project is to understand and implement how Gaussian Process (GP) based models can be used to learn such representations.

Keywords : Representation Learning, Gaussian Processes, Graphical Models, Variational Inference

Introduction

Feature learning or representation learning is a set of techniques that can automatically discover representations (often lower dimension) that can improve the performance on downstream tasks such as classification or regression (Wikipedia contributors 2018). Representation learning often requires learning a distribution over the latent variables, given the data, so that new data samples can be generated by sampling from the latent space. Representation learning can be done in both supervised setting (for example extracting last layer representations from a CNN trained on image classification) and unsupervised setting (such as an Auto encoder or RBM), which is trained layer by layer (Goodfellow, Bengio, and Courville 2016).

Representation learning has gained importance recently and empirical studies show performance gains in various across that include computer vision, signal processing and natural language processing (Bengio, Courville, and Vincent 2013).

One of the most common representation learning models is the Restricted Boltzmann Machine, which is a generative model and learns the distribution over the latent space given the input data. It can be used to generate new data points by sampling from the distribution in latent space and multiplying with the weights. Another common model is the Autoencoder, which is a two layer neural network with shared weights that aims to learn the latent representation that minimizes the reconstruction error (Salakhutdinov, Mnih, and Hinton 2007). It was observed that generating latent representations from an autoencoder may lead to representations that are not restricted and may not give any improvement in a downstream class. Variational Auto encoders tackle this by assuming a Gaussian prior over the latent space, and allows sampling from the latent space (Kingma and Welling 2013).

More recently Gaussian process have been used to learn latent representations. This project explores the use of Deep Gaussian process latent variable models for representation learning across three datasets. We see from experiments that the deeper models can learn much better representations, and these representations can be used to get better performance in downstream tasks like classification.

Background

Gaussian Processes

Gaussian processes are stochastic process over a space of functions that assume a Gaussian prior over the functions, defined by a mean function and a kernel function. Gaussian processes are non parametric models that try to learn a posterior over the space of all possible functions defined by the mean and the kernel functions that best describe the data (Wikipedia contributors 2019). Since there can be infinitely many inputs for continuous functions, the GPs are usually constructed from a finite set of observed points, and the posterior at each point gives a value as well as the uncertainty in the estimation, and is also a conditional normal distribution (Katherine Bailey 2016). Since Gaussian processes are defined over a function space, they can be very powerful and can learn a family of functions that can fit any arbitrary data distribution.

Sparse Gaussian Processes

Even though Gaussian Processes are very powerful the solution to the Gaussian processes however can be unfeasible when the dataset is very large as it involves a cubic time inversion operation applied to the kernel matrix. It was observed that reasonable estimations to the posterior can be made by simply using a random subset of data points (inducing points). Further improvements can be observed if these inducing points are chosen cleverly through variational

inference giving rise to sparse Gaussian processes (Snelson and Ghahramani 2006). (Sheth, Wang, and Khardon 2015) attempt to further generalize sparse Gaussian processes and include different observation likelihoods and thus more complicated models, and propose a faster fixed point approximation for training these models.

Related Work

As Gaussian processes and sparse gaussian processes show impressive results and scalability with the dataset size, they have also been used to model latent representations from the data.

Lawrence et. al in (Titsias and Lawrence 2004) formulate a Gaussian Process Latent Variable Model (GPLVM) probabilistic version of PCA where the data is assumed to be generated by a Gaussian process from the latent space. Instead of integrating out the latent variables, the authors integrate out the parameters in the likelihood, and optimize for the latent variables to get an optimal latent representation. The amount of non-linearity can be controlled by the choice of the family of kernel functions, and it was shown that a linear kernel function leads to the PCA solution.

Titsias et. al in (Titsias and Lawrence 2010) use variational inference using the mean field approximation to vatiationally integrate out the input variables and optimize the lower bound on the marginal likelihood of the latent variables (Bayesian GPLVM). Using variational inference provides a Bayesian training procedure that is robust to overfitting and can learn the dimensionality of the latent space using the automatic relevance determination squared exponential kernel (ARD SE) kernel.

As an extension to the Bayesian GPLVM Damianou et. al in (Damianou and Lawrence 2013) propose a hierarchical model where each layer is a Gaussian process, and show that such a deep model can learn increasingly complicated functions. Such deep Gaussian processes can be used in multiple output scenarios as well as for representation learning, and show better results as compared to a single Gaussian process model. The deep model is trained layer by layer.

There are various libraries that implement Gaussian processes such as PyMC (Salvatier, Wiecki, and Fonnesbeck 2016), which uses Monte Carlo approximations, which can lead to better approximations but take a long time to converge. Scikit-learn (Pedregosa et al. 2011) also implements Gaussian processes but the implementation is limited only to regression and classification scenarios. GPy is another library that implements Gaussian processes in python and has a Bayesian GPLVM implementation. GPFlow (Matthews et al. 2017) is the extension of GPy library ans uses a Tensorflow backend that allows for faster training. For this project GPFlow implementation of the Bayesian GPLVM was used, and extended to create a deep architecture. The details of the implementation and the experimental results are shown below.

Experiments

The ability of the Bayesian GPLVM to learn latent representations was explored in this project. The latent representations from three different datasets were learned using the single layer Bayesian GPLVM and the deep GPLVM models, with different parameters and the results are shown below. The evaluation was done based on downstream classification accuracy and the reconstructions from the latent representations.

Datasets

Three datasets were used in the experiments:

- **3 Phase Oil Flow Dataset:** The oil flow dataset (Neil Lawrence) consists of data from pipelines that can carry oil, water and gas. The flow of in the pipeline can be one of the three categories - horizontally stratified, nested annular or homogeneous mixture flow. The dataset consists of 12 features and 1000 training examples and the task is to predict the type of flow given the observations.
- **MNIST Handwritten Digits:** The MNIST dataset (Yann LeCun, Corinna Cortes, Christopher J.C. Burges) consists of 60,000 training images and 10,000 test images of handwritten digits from 0 to 9. The images are 28x28 pixels and consist of corresponding labels. The dataset is used for the digit recognition task, which is a classification problem with 10 classes.
- **Frey Faces Dataset:** The Frey Faces dataset (Sam Roweis) consists of almost 2000 images of the face of Brendan Frey. Each of the images only consists of one face in different orientations and are 20x28 pixels in size. The dataset does not contain any labels and is used for unsupervised learning tasks.

Training

The single layer Bayesian GPLVM implementation from the GPFlow library was used to learn the latent variable representation on the dataset. The latent representation was initialized by using PCA components from the input data, and optimal representations were found by opfimization over many iterations.

The deep GPLVM was built by stacking the single layer GPLVMs and training was done layer by layer, from the highest layer (closest to the input data) to the lowest layer (the latent representation). At each layer the hidden representations were initialized with the PCA components of the representations learned so far. The dimensionality of the latent space was decreased successively across the different layers.

In both cases the RBF kernel was used and training was done based on a single Tesla K80 GPU provided by Google Colaboratory notebooks (Google Contributors).

Optimization

The GPflow library consists of various different optimizers to choose from, and initial experiments were performed to compare the default Scipy Optimizer, which provides Tensorflow wrapper to the optimizer module in scipy, and the Adam optimizer. However, no significant difference was found in the latent representations, and therefore, the Scipy Optimizer was used, as it was more stable, at least for the

datasets considered here. The optimizer minimizes the negative ELBO for the Bayesian GPLVM and therefore, indirectly minimizes the KL-divergence between the approximating distribution and the true posterior distribution over the latent space.

The code is provided along with the report.

Results

Figure 1 shows the latent representations from a single layer GPLVM, with 20 inducing points on the oil flow dataset. As we can see from the plots, the class separation is much clearer using the GPLVM as compared to PCA.

The boxplot in figure 2 compares the downstream classification accuracy between using the raw data, the PCA components, and the single layer and 2 layer models. We can see that the deeper models learn representations that can be meaningful for downstream tasks.

We repeat similar experiments on a subset of the MNIST dataset that contains 50 images from each of the classes between 0 to 4. Figure 3 shows the scatter plots of the representations learned on MNIST dataset. We can see better class separation in the deep GPLVM model.

Figure 4 and 5 show the latent representations and reconstructions from the single layer GPLVM and the 2 layer deep GPLVM model.

The boxplot in figure 8 compares the downstream classification accuracy of the models across various random splits of the dataset. as we can see, the latent representations learned from GPLVM outperform representations learned from PCA, even when we have few data points.

Finally figures 6 and 7 show the learned length scales of different models that have been used above that show the feature importance in the latent space.

Finally we learn a 3 dimensional latent representation from the Frey Faces dataset, and figures 9 and 10 show the reconstructions obtained from the 2 layer, and 3 layer models respectively.

We also explore what each of the features mean and we see that the 3 layer deep model can learn latent features that encode information about the smile, and orientation of the face as shown in figure 11.

Conclusion

As we can see, Gaussian process LVMs can be used to learn non-linear mappings to a lower dimensional latent space that is compact and can encode useful information which can be important to downstream tasks such as classification. The Bayesian GPLVMs and the deep model extensions are Generative models, and it was observed that they can suffer from mode collapse (where all the reconstructions become the same or correspond to one mode). This problem is more prominent in deeper architectures, and therefore it is important to successively reduce the dimensionality from one layer to the next, and to not overfit too much on any one layer to avoid this. It was also observed that the number of inducing points used does not affect the latent representations and reconstructions too much in our case. However, choosing very few inducing points can lead to poorer performance in larger datasets.

Future Work

Although we see good reconstructions on images in this project, any spatial information is unaccounted for and each pixel is considered an independent input. Convolutional kernel based deep Gaussian processes (Kumar et al. 2018) adresses this problem and should be explored as futire work. The different layers were trained one at a time in a greedy manner. Better performance may be possible when training the entire deep GPLVM together dropping the independence assumption between the layers. Salimbeni and Deisenroth in (Salimbeni and Deisenroth 2017) propose a method that allows training of deeper GP models, and should be explored as future work.

Acknowledgements

I would like to thank professor Roni Khardon for giving me the opportunity to work on this project and for various insights in the class that helped in the completion the project.

References

[Bengio, Courville, and Vincent 2013] Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35(8):1798–1828.

[Damianou and Lawrence 2013] Damianou, A., and Lawrence, N. 2013. Deep gaussian processes. In *Artificial Intelligence and Statistics*, 207–215.

[Goodfellow, Bengio, and Courville 2016] Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.

[Google Contributors] Google Contributors. Google colaboratory.

[Katherine Bailey 2016] Katherine Bailey. 2016. Gaussian processes for dummies.

[Kingma and Welling 2013] Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[Kumar et al. 2018] Kumar, V.; Singh, V.; Srijith, P.; and Damianou, A. 2018. Deep gaussian processes with convolutional kernels. *arXiv preprint arXiv:1806.01655*.

[Matthews et al. 2017] Matthews, A. G. d. G.; van der Wilk, M.; Nickson, T.; Fujii, K.; Boukouvalas, A.; León-Villagrá, P.; Ghahramani, Z.; and Hensman, J. 2017. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research* 18(40):1–6.

[Neil Lawrence] Neil Lawrence . 3 phase oil data.

[Pedregosa et al. 2011] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

[Salakhutdinov, Mnih, and Hinton 2007] Salakhutdinov, R.; Mnih, A.; and Hinton, G. 2007. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, 791–798. ACM.

[Salimbeni and Deisenroth 2017] Salimbeni, H., and Deisenroth, M. 2017. Doubly stochastic variational inference for deep gaussian

processes. In *Advances in Neural Information Processing Systems*, 4588–4599.

[Salvatier, Wiecki, and Fonnesbeck 2016] Salvatier, J.; Wiecki, T. V.; and Fonnesbeck, C. 2016. Probabilistic programming in python using pymc3. *PeerJ Computer Science* 2:e55.

[Sam Roweis] Sam Roweis. Frey faces dataset.

[Sheth, Wang, and Khardon 2015] Sheth, R.; Wang, Y.; and Khardon, R. 2015. Sparse variational inference for generalized gp models. In *International Conference on Machine Learning*, 1302–1311.

[Snelson and Ghahramani 2006] Snelson, E., and Ghahramani, Z. 2006. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, 1257–1264.

[Titsias and Lawrence 2004] Titsias, M. K., and Lawrence, N. D. 2004. Gaussian process latent variable models for visualisation of high dimensional data. In *Adv. in Neural Inf. Proc. Sys.* Citeseer.

[Titsias and Lawrence 2010] Titsias, M., and Lawrence, N. D. 2010. Bayesian gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 844–851.

[Wikipedia contributors 2018] Wikipedia contributors. 2018. Feature learning — Wikipedia, the free encyclopedia. [Online; accessed 25-April-2019].

[Wikipedia contributors 2019] Wikipedia contributors. 2019. Gaussian process — Wikipedia, the free encyclopedia. [Online; accessed 26-April-2019].

[Yann LeCun, Corinna Cortes, Christopher J.C. Burges] Yann LeCun, Corinna Cortes, Christopher J.C. Burges. The mnist database of handwritten digits.

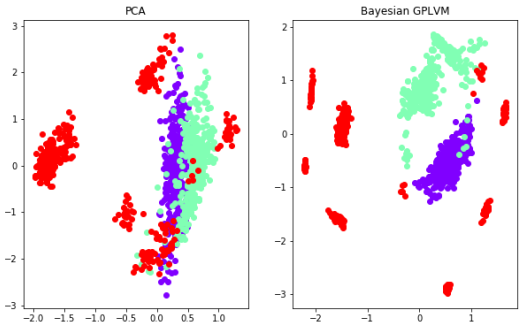


Figure 1: Scatter plot of the two most dominant PCA and 2 layer GPLVM latent features on oil flow dataset

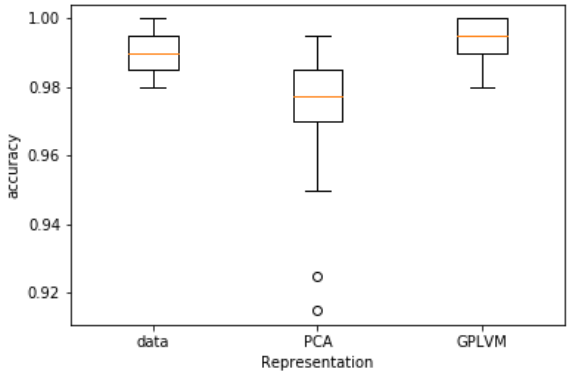


Figure 2: Boxplots of classification accuracy across 500 different random 80-20 training and testing splits of the Oil-flow dataset, using representations from different modelst

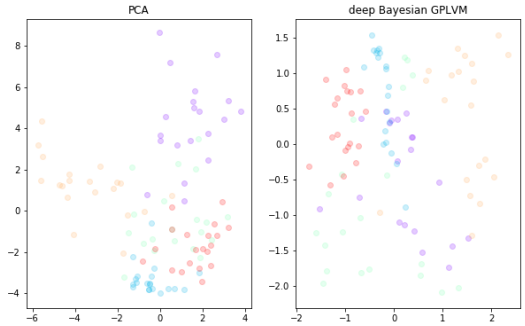
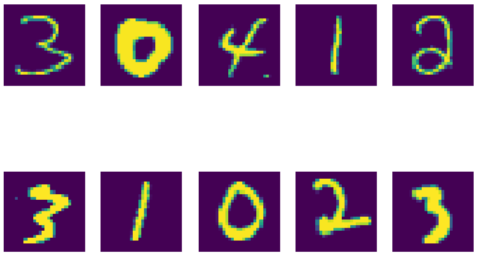
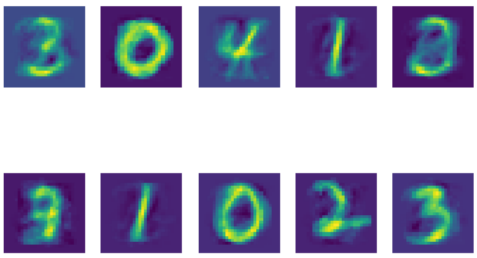


Figure 3: Scatter plot of the two most dominant PCA and 2 layer GPLVM latent features on MNIST dataset



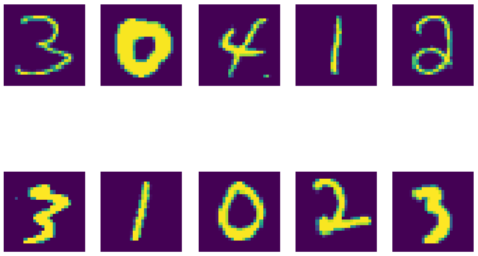
Original images from MNIST dataset



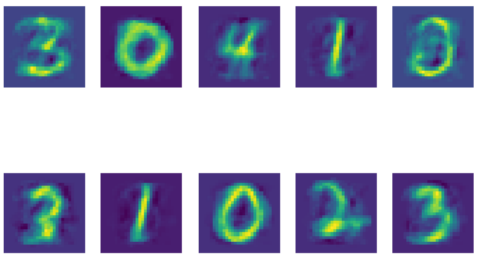
Reconstructions from the single layer GPLVM

Corresponding 5 dimensional latent representation form single layer GPLVM

Figure 4: Reconstructions and representations for a sample of images using one GPLVM on the MNIST dataset



Original images from MNIST dataset



Reconstructions from the two layer deep GPLVM

Corresponding 5 dimensional latent representation form two layer deep GPLVM

Figure 5: Reconstructions and representations for a sample of images using 2 layer deep GPLVM on the MNIST dataset

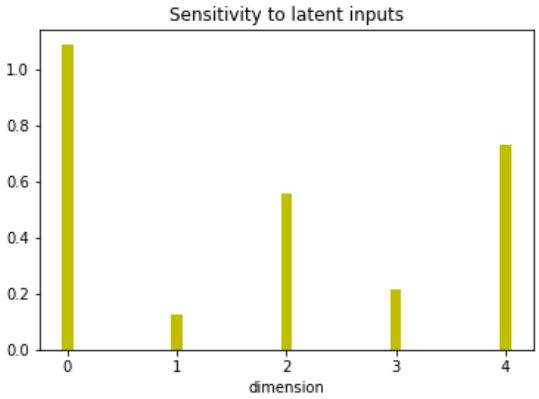


Figure 6: Length scales for the one layer GPLVM model

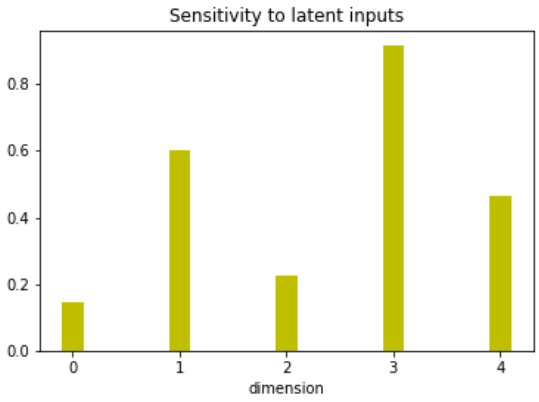


Figure 7: Length scales for the two layer deep GPLVM model

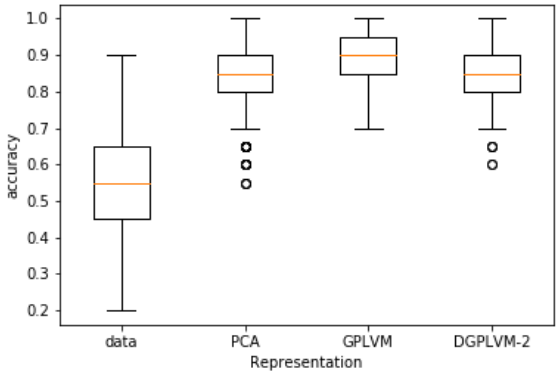
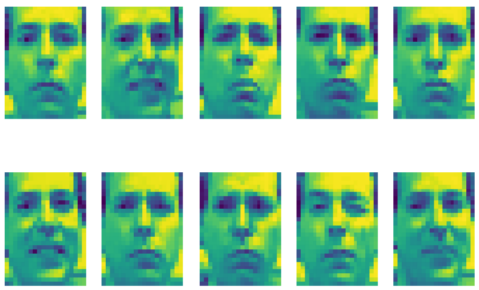
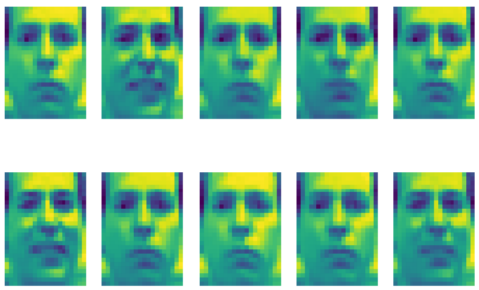


Figure 8: Boxplots of classification accuracy across 500 different random 80-20 training and testing splits of the MNIST dataset, using representations from different models



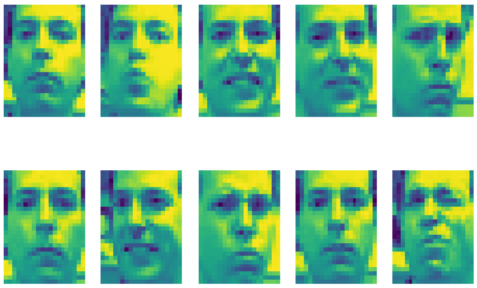
Original images from Frey Faces dataset



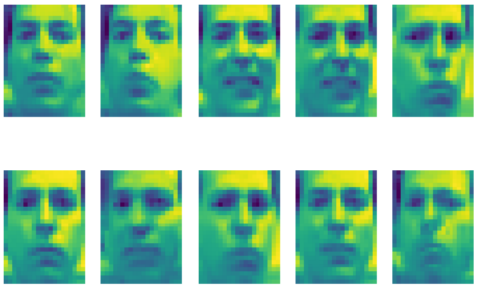
Reconstructions from the two layer deep GPLVM

Corresponding 3 dimensional latent representation form two layer deep GPLVM

Figure 9: Reconstructions and representations for a sample of images using 2 layer deep GPLVM on the Frey Faces dataset



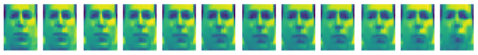
Original images from Frey Faces dataset



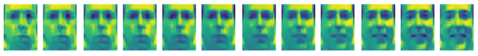
Reconstructions from the three layer deep GPLVM

Corresponding 3 dimensional latent representation form three layer deep GPLVM

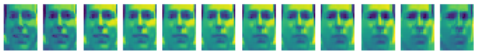
Figure 10: Reconstructions and representations for a sample of images using 3 layer deep GPLVM on the Frey Faces dataset



Samples generated by changing the first feature



Samples generated by changing the second feature



Samples generated by changing the third feature

Figure 11: Understanding what each of the three latent features from the 3 layer deep GPLVM mean