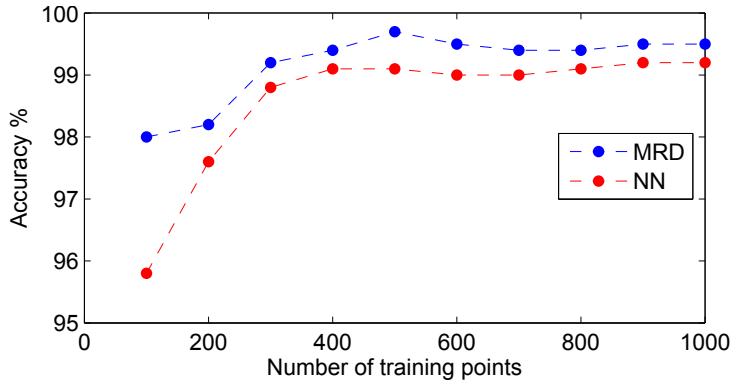


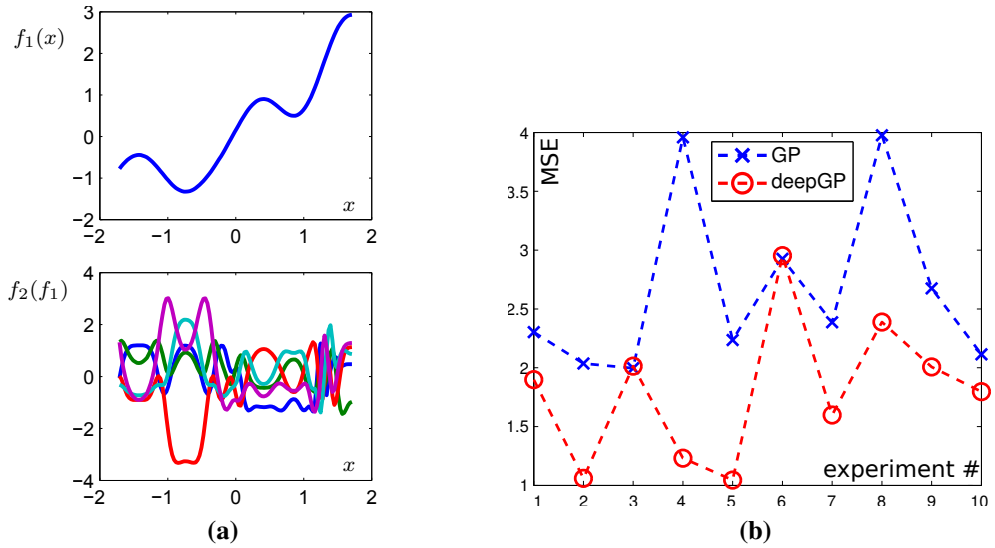
Fig. 5.11: Accuracy obtained after testing MRD and NN on the full test set of the 'oil' dataset.

being represented by the raw values of the 60×80 pixels around the lips, as can be seen in figure 5.13. Thus, a single instance of the video modality of this dataset is a $115200-$ dimensional vector.

Data Exploration

Depending on the desired predictive or exploratory task, different subsets of the data can be split across different views. To explore the connections and commonalities in the information encoded in different subjects, letters and type of signal (video or audio), we first performed data exploration by considering the following generic setting: we created a dataset where the modalities were split across all subjects and across type of signal. We only considered 8 of the subjects. Thus, we ended up with 16 different modalities, where modalities $i, i+1$ contained the video and audio signal respectively for the $i-$th subject. The alignment was therefore made with respect to the different letters. We used all three available trials but letters "B", "M" and "T" were left out of the training set completely. For each modality, we thus had 69 rows (23 letters $\times$ 3 trials). The split across instances and modalities is summarised in Table 5.2. In the test set, each modality had only 9 rows (3 letters $\times$ 3 trials). Notice that this is a rather extreme scenario: the number of training instances is only 4.3 times larger than the number of modalities. We applied MRD to reveal the strength of commonality between signal corresponding to different subjects and to different recording type (video/audio). The visualisation of the ARD weights can be seen in figure 5.14.

This figure shows that similar weights are typically found for modalities $1, 3, 5, ...$, i.e. for the ones that correspond to the video signal. This means that if, for example, one would like to predict the lip movements in a test scenario, the other pieces of



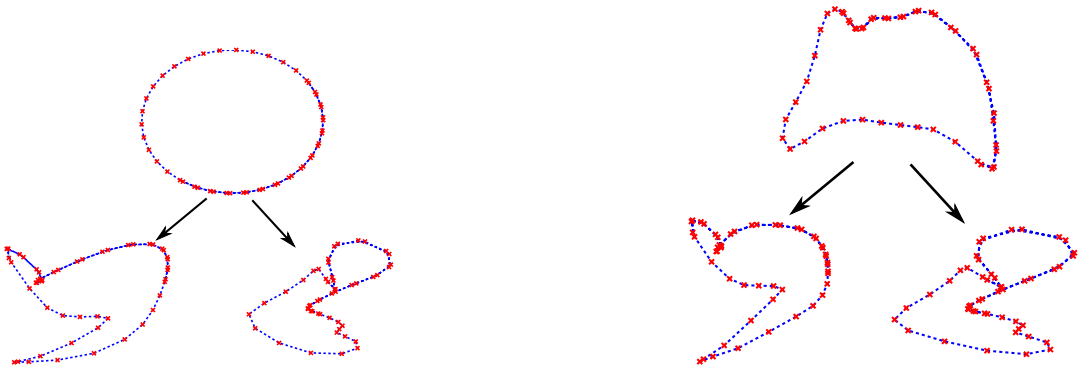
(a) **(b)**

Fig. 6.8: Figure (a) shows the toy data created for the regression experiment. The top plot shows the (hidden) warping function and bottom plot shows the final (observed) output. Figure (b) shows the results obtained over each experiment repetition.

these long range dependencies, similarly to the above step function demonstration. Another way of thinking of data like this is as a nonlinear warping of the input space to the GP. Because this type of deep GP only contains one hidden layer, it is identical to the the dynamical variational GP-LVM [Damianou et al., 2011]. With the deep GP models described in this chapter the aim is to provide a more complex deep hierarchy, but still learn the underlying representation correctly. To this end, a standard GP (1 layer less than the actual process that generated the data) and a deep GP with two hidden layers (1 layer more than the actual generating process) were applied. The experiment was repeated 10 times, each time obtaining different samples from the simulated warped process and different random training splits. The results show that the deep GP predicted better the unseen data, as can be seen in figure 6.8(b). The results, therefore, suggest that the deep model can at the same time be flexible enough to model difficult data as well as robust, when modelling data that is less complex than that representable by the hierarchy. It can be presumed that these characteristics are due to the Bayesian learning approach that deals with capacity control automatically.

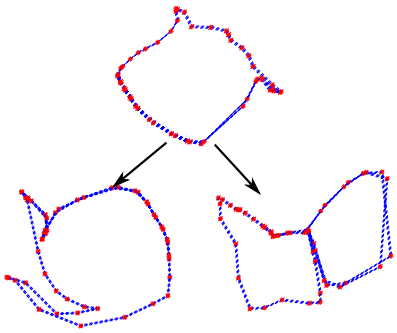
Toy Manifold Learning Problem

As a final demonstration on toy data, a hierarchy of signals was created by sampling from a three-level stack of GPs. Figure 6.9 (a) depicts the true hierarchy: from the top latent layer two intermediate latent signals are generated. These, in turn, together

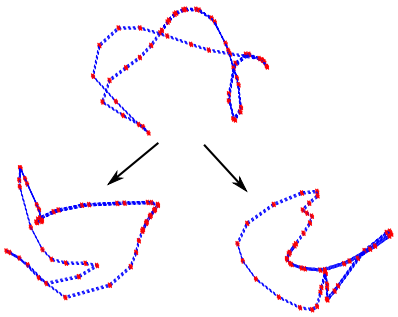


(a) Real data.

(b) Deep GP reconstruction.



(c) Stacked Isomap reconstruction.



(d) Stacked PCA reconstruction.

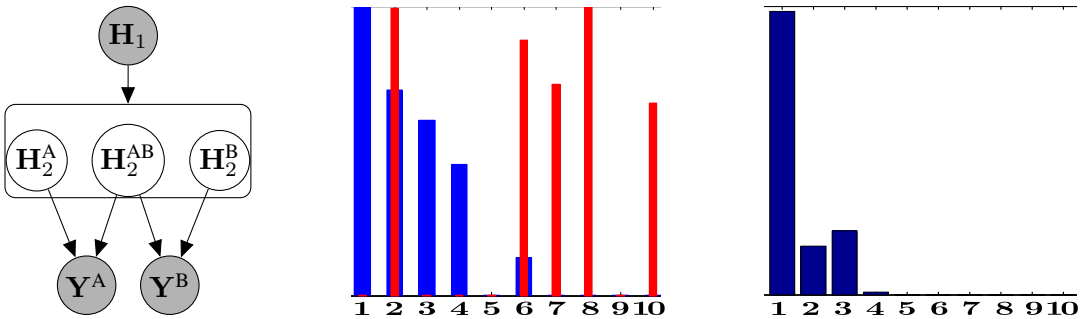
Fig. 6.9: Attempts to reconstruct the real data (fig. (a)) with our model (b), stacked Isomap (c) and stacked PCA (d). Our model can also find the correct dimensionalities automatically.

generate 10-dimensional observations (not depicted) through sampling of another GP. These observations are then used to train the following models: a deep GP, a simple stacked Isomap [Tenenbaum et al., 2000] and a stacked PCA method, the results of which are shown in figures 6.9 (b, c, d) respectively. From these models, only the deep GP marginalises the latent spaces and, in contrast to the other two, it is not given any information about the dimensionality of each true signal in the hierarchy; instead, this is learned automatically through ARD. As can be seen in figure 6.9, the deep GP finds the correct dimensionality for each hidden layer, but it also discovers latent signals which are closer to the real ones. This result is encouraging, as it indicates that the model can recover the ground truth when samples from it are taken, and gives confidence in the variational learning procedure.

Fig. 6.11: The nearest neighbour class separation test on a deep GP model with depth 5. This plot shows the top layer's latent space projection on its two principal dimensions (5 and 6). The output images corresponding to the top layer's training inputs are superimposed on the plot (some instances resulting in occlusions were removed). This demonstrates the robust separation learned by the deep model in a completely unsupervised manner (i.e. no class labels were given to it). It is interesting to notice the digits on the border of the clusters. For exampe, the zeros that are close to the cluster of ones are very elongated and those that are further are very round.

of the other two encodes information for each of the two interacting subjects. Our method is not constrained to two dimensional spaces, so for comparison we plot two-dimensional projections of the dominant dimensions of each subspace in figure 6.14 (a,b,c). The similarity of the latent spaces is obvious. In contrast to Lawrence and Moore [2007], we did not have to constrain the latent space with dynamics in order to obtain results of good quality.

Further, we can sample from these spaces to see what kind of information they encode. Indeed, we observed that the top layer generates outputs which correspond to different variations of the whole sequence, while when sampling from the first layer we obtain outputs which only differ in a small subset of the output dimensions, e.g. those corresponding to the subject's hand.



(a) A multi-view deep GP model.

(b) ARD weights for the two views of the bottom layer.

(c) ARD weights for the top latent layer.

Fig. 6.13: Figure (a) shows the (multi-view) deep GP model employed. Figure (b) shows the ARD weights for the bottom layer's mappings f_2^{A} (blue/wider bins) and f_2^{B} (red/thinner bins). Dimensions 2 and 6 form the shared space. Figure (c) shows the ARD weights for the top layer's mappings, f_1.

Figure 6.15 depicts the results by showing the projection on the most dominant dimensions of the top layer's latent space. As can be seen, although the models were not given the temporal information for the video data and the label information for the oil flow data, they managed to discover latent spaces that encapsulate this information naturally. In particular, concerning the oil flow data, the nearest neighbour error in the projection is 0, meaning that all points are clustered very well in relation to their label. This figure can be compared to the unsupervised learning case of figure 3.5a. The ARD weights were very similar to those of figure 3.4. Notice that the latent points represented as red crosses form an "L" shape with those of the class corresponding to green circles in a third dimension (not visualised), perpendicular to the page. Figure 6.15c shows the Frey faces data outputs centered on their corresponding latent locations (only a small subset is shown due to removing overlaps). Many aspects of this high-dimensional data was captured on only two dimensions. Firstly, the outliers (e.g. top image and low, far left) were placed away from the rest of the points. Other quite peculiar grimaces are also clustered together (winking, tongue out etc). Secondly, we see multiple levels of separation: moving top-down on the $y-$axis, the faces gradually change rotation from looking on the (subject's) right to the left. Further, "happy" faces are placed on the left and "sad" and then "angry" faces are placed on the right.

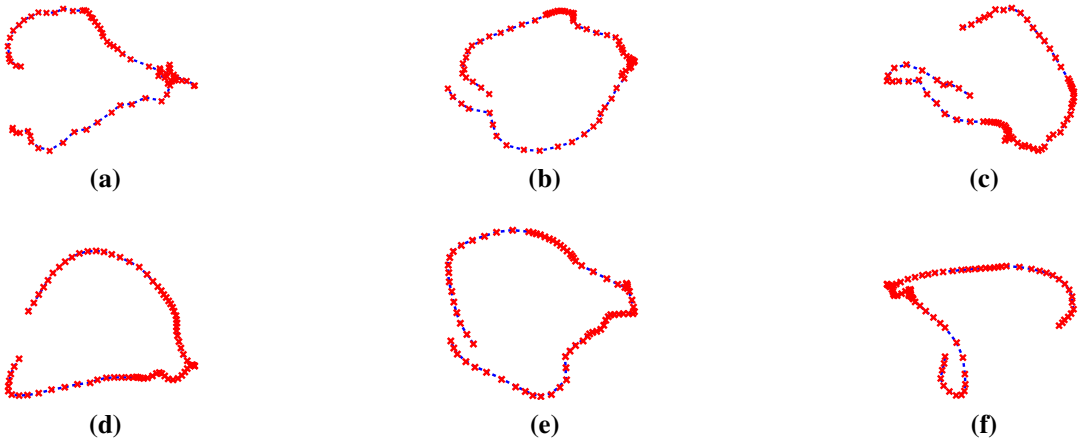


Fig. 6.14: Up (a,b,c): projections of the latent spaces discovered by our model, Down (d,e,f): the full latent space learned for the model of Lawrence and Moore [2007].

Evaluating the Compression Quantitatively

To evaluate the quality of the compression achieved by the autoencoders, we applied the models on data associated with labels (not given to the model) and used the discovered latent space (means of the variational distributions) as features for a discriminative classifier. We used the oil flow dataset (1000 training and 1000 test instances) and the USPS digit data subset that was considered in Section 6.3.2 (150 training and 150 test examples). We compared models M_1, M_2 and M_3 in both, the "shallow" and the deep setting. However, the deeper models produced similar results to the shallow ones but with an extra optimisation burden. Therefore, in the rest of the analysis we restrict our attention to the shallow models.

To increase the reliability of the results we tried two different classifiers: a vanilla support vector machine (SVM) and multiple logistic regression (MLR). The results are summarised in Table 6.1. As can be seen, the autoencoders result in better performance. On the other hand, optimising an autoencoder during the training phase is, in general, more challenging due to the increased number of parameters. In a few cases we needed to restart the optimisation due to getting stuck in local minima, although for the unsupervised learning model this was generally not needed. Another observation from our experiments is that the autoencoder M_2 requires much more iterations to converge, possibly because of the correlated structure in the posterior. In conclusion, the important result to keep from these experiments is that the autoencoders perform at least as well as the unsupervised equivalent but are much faster at test time.