Non-Gaussian and Non-linear Latent Variable Models

In factor analysis, the m latent variables, $z_1, \ldots, z_m$, have independent Gaussian distributions, and the relationship of the obsderved variables, $x_1, \ldots, x_p$, to z is assumed to be linear.

We could change either or both of these assumptions:

Independent Component Analysis (ICA) keeps the linear relationship of x to z, and $z_1, \ldots, z_m$ are still independent, but it assumes that each z_k has *anything but* a Gaussian distribution.

With this change, the non-uniqueness of factor analysis (when $m > 1$) goes away. It turns out that spherical Gaussian distributions are the *only* ones that are rotationally symmetrical and have independent components.

Nonlinear latent variable models may (or may not) keep the Gaussian distribution for z, but they assume that the relationship of x to z may be nonlinear.

Challenges of Nonlinear Latent Variable Modeling

Care is needed to avoid overfitting. Allowing arbitrarily peculiar functions from z to x would fit the training data well, but not result in good predictions, nor in valid insight into the nature of the data.

Bayesian or penalized maximum likelihood methods could be used.

Inverting the model may be computationally difficult. The model may directly specify how x relates to z. But if we observe a new x, we may want to infer what z produced it (or a distribution over possible values for z). This can be difficult, whereas for factor analysis, the distribution of z given x is Gaussian, with a mean that is a linear function of x.

We may need to use Markov chain Monte Carlo methods. Alternatively, we might train a "recognition" model for this task along with the main "generative" model.

Estimating the parameters can be computationally difficult. If maximum likelihood is used, there may be many local optima. Markov chain Monte Carlo methods may take a long time to converge.

Auto-Encoder Neural Networks

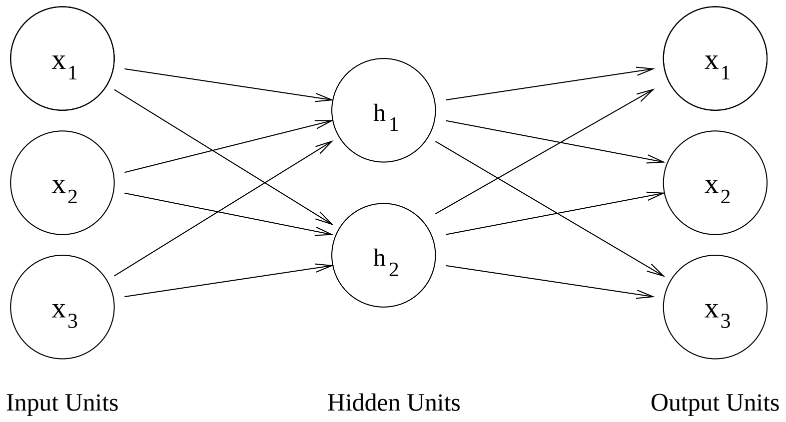
One approach to these problems is to train an *auto-encoder*, which combines the recognition and generative models.

We choose some supervised learning procedure — eg, a multilayer perceptron network — but rather than have it predict some response y from x, we instead train it to predict x from x,

Of course, this is trivially easy if no constraint is put on how the predictions can be done. We need to put a"bottleneck" in the model that prevents it from just predicting x perfectly by saying it's equal to x.

A Neural Network for Principal Component Analysis

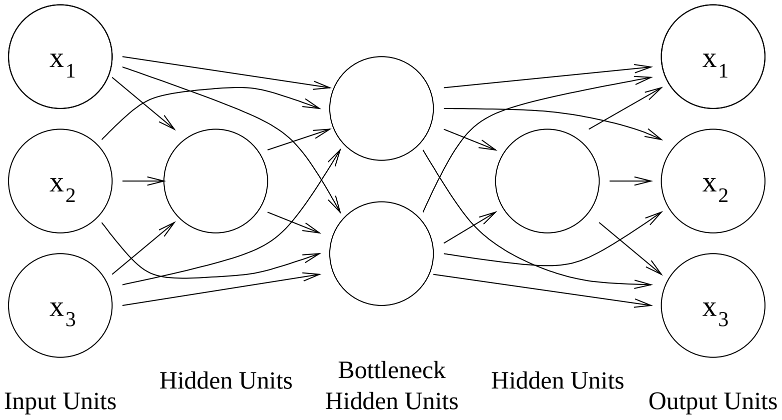
Here's a simple auto-encoder network that finds vectors that span the space of the first m principle components:



We train this network to minimize the sum of the squared reconstruction errors. The hidden units (which have identity activation functions) will compute $E(z|x)$.

Making the Auto-Encoder Nonlinear

We can change this network so that the mappings to and from the bottleneck are nonlinear:



Here, I've put in direct connections from the inputs to the middle hidden layer (the bottleneck) and from the bottleneck to the output units. There's only one hidden unit in the extra hidden layers, but usually one would have more.

Constrained Linear Dimensionality Reduction

Another direction for modifying PCA is to introduce constraints.

Often, we think that data is a non-negative combination of some non-negative patterns. For example, the spectrum of a serum sample is the sum of contributions from the various molecules in the serum. The points in each molecule's spectrum are non-negative, and the amounts of each molecule in the serum are also non-negative.

Non-negative Matrix Factorization finds such a decomposition. It finds a way of writing the $n \times p$ matrix, X, of observed data in the form

$$X = WH + E$$

where W is $n \times m$, H is $m \times p$, and E is $n \times p$. W and H are non-negative. The matrix E represents "noise" that isn't accounted for by the factorization. We might aim to minimize the sum of squares of values in E.

This decomposition isn't unique, but may nevertheless provide insight into the data. In contrast, results of PCA may be hard to interpret, since positive and negative components can cancel.