COMP9844: Neural Networks

2. Autoencoder Networks

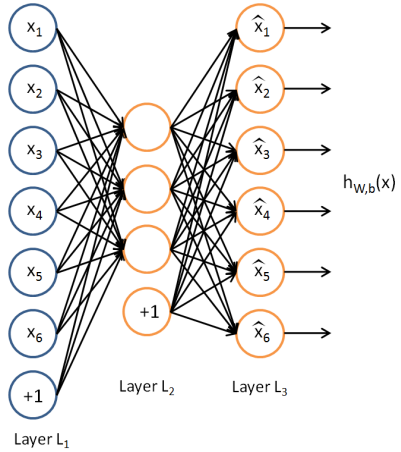
Outline

- Autoencoder networks
- Denoising Autoencoders
- Stacked Autoencoders
- Sparse Autoencoders

Autoencoder networks

- Autoencoders are used to capture structure in data, using unsupervised learning
- Data is provided as input, and the output of the network tries to reconstruct the input
- Learning is performed using backpropagation or related methods
- The network captures a reduced representation of inputs
- Useful for pre-training a network, improving learning and allowing greater depth

Autoencoder networks



Autoencoder networks

- Autoencoders are a multi-layer neural network with a specific topology
- The target output of the network is set to the input
- The aim of training is to minimise the error of reconstruction
- Often a reduced set of hidden units is used, creating an information bottleneck

Autoencoder networks

- Weights between the input and hidden layer are often tied with weights between the hidden layer and output
- Given an input vector $\mathbf{x} \in [0,1]^d$, hidden unit and output activations are calculated as:

$$\mathbf{y} = \varphi(\mathbf{W}\mathbf{x} + \mathbf{b})$$

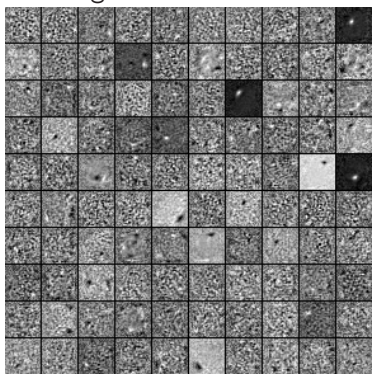
$$\mathbf{z} = \varphi(\mathbf{W}'\mathbf{y} + \mathbf{b}')$$

- Reconstruction error can be calculated using a number of methods, including squared error:

$$E = \frac{1}{2}\|\mathbf{z} - \mathbf{x}\|^2$$

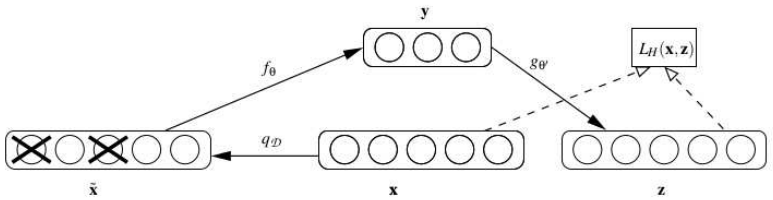
Implementation

- A description of autoencoder implementation is given at: `http://deeplearning.net/tutorial/dA.html`



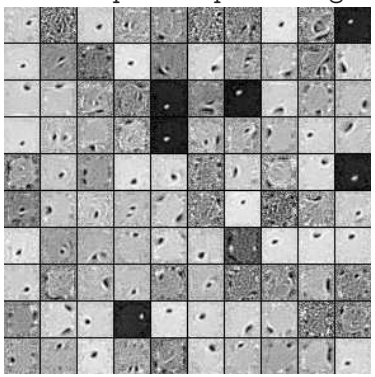
De-noising Autoencoders

- To avoid overfitting, and to encourage learning of structure instead of noise, de-noising autoencoders are used
- Method: add noise to inputs used for learning



Implementation

- Further reading, and a description of denoising autoencoder implementation is given at: `http://deeplearning.net/tutorial/dA.html#denoising`



Stacked Autoencoders

- To initialise a deep network based on unsupervised learning, autoencoders can be stacked
- Each layer is trained in turn, and used as input for the next layer
- This provides an effective initialisation of the network, before supervised learning is used
- Further reading: `http://deeplearning.net/tutorial/SdA.html`

Sparse Autoencoders

- We want to encourage the network to discover structure of the input, instead of capturing noise, or learning a trivial mapping between inputs and outputs
- Fewer hidden nodes can encourage feature discovery (bottleneck), however with a larger number of hidden nodes we can improve discovery of structure through encouraging sparsity on hidden units

Sparse Autoencoders

- To encourage sparse representation, a penalty term is added to the error function, to penalise when hidden units are active frequently, for example:

$$\sum_j KL(\rho \| \overline{y_j})$$

- This is based on a measure of the average activation of each hidden unit, which we want to be small, such as $\rho = 0.05$. The Kullback-Liebler divergence is minimised when $\overline{y_j} = \rho$
- This constraint is satisfied when the network captures a sparse coding
- Further reading: `http://deeplearning.stanford.edu/wiki/index.php/Autoe`