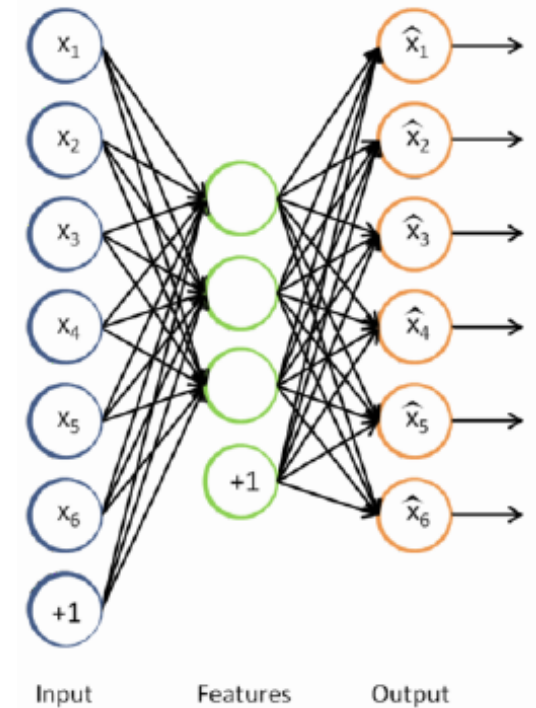


CS 615 – Deep Learning

Auto-Encoders

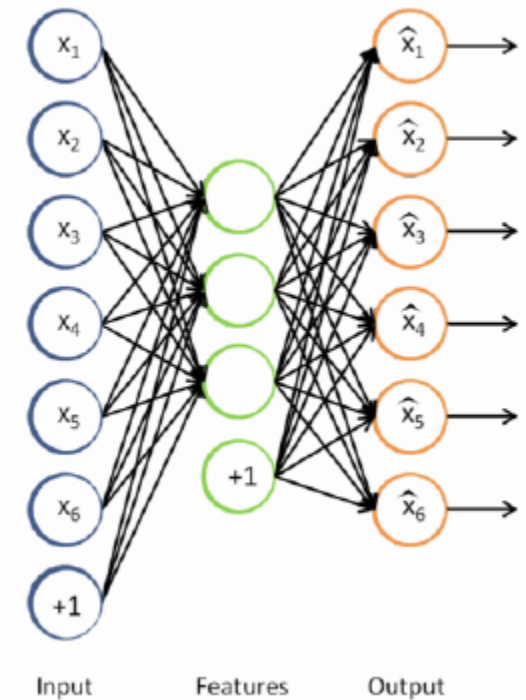
Auto-Encoders

- As the name implies, auto encoders can be useful for encoding (and decoding) data.
- The weights from the input to the hidden layer form the encoders
- The weights from the hidden layer to the output layer form the decoders.
- The output is the same as the input
 - Therefore this is like a *self-supervised* learning algorithm.



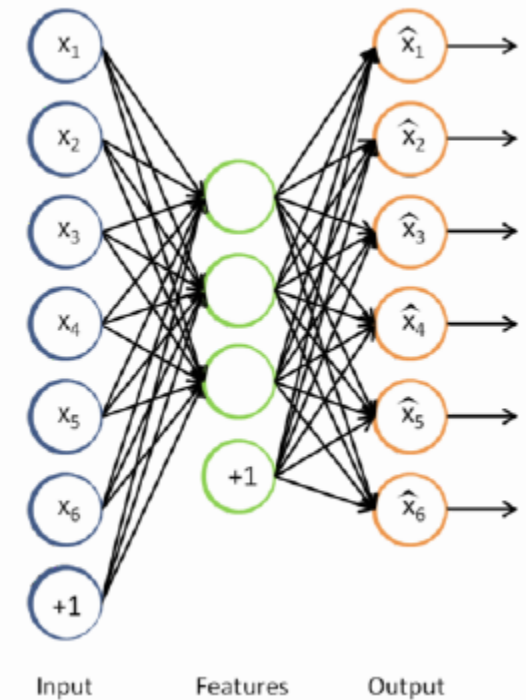
Auto-Encoder

- The basic process to design an auto-encoder is:
 1. Take the input and add a bias node (optional)
 2. Choose the hidden layer size to be less than the input size (lossy encoding)
 3. The output layer should be the same size as the input (minus the bias node)
 4. Train this auto-encoder using the data as the desired input and output values.



Auto-Encoder

- Another common use of Auto-Encoders is for *dimensionality reduction*
- There are times that we want to reduce the number of features, without losing the “important information”.
- While there are several approaches to this, if we create an auto-encoder where the number of nodes in the hidden layer is less than the input layer, then the encoder portion can be thought of as a dimensionality reduction machine.



Auto-Denoiser

- Auto-encoders can also be used for de-noising!
 1. Take the input, add some noise to it, and add a bias node
 2. Choose the hidden layer size to be less than the input size
 3. The output layer should be the same size as the input (minus the bias node)
 4. Train this auto-encoder using the uncorrupted data as the desired output values.

Stacked auto-encoders

- We can also use auto-encoders (and even several of them!) as initial stages to a deep learning.
- Do auto-encoding to learn the first layer.
- Then use the output of the learned hidden layer to training another auto-encoder.
- Etc..
- Then do supervised training on last layer using some traditional learning.
- Then do supervised training on whole network to fine tune the weights

Stacked auto-encoders

