# Sparse Autoencoders

Autoencoders are neural networks that try to learn the identity function. Sparsity constraints force the learned representation to be nontrivial.

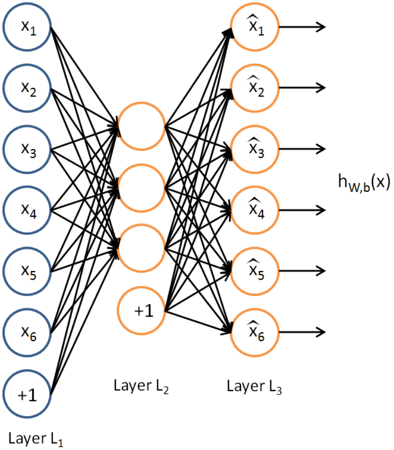


Figure: Autoencoder

# Cost Function

Let the hidden layer have $p$ units, and let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable activation function. The neural network is parameterized by terms weights $W^{(1)} \in \mathbb{R}^{p \times n}$ and $W^{(2)} \in \mathbb{R}^{n \times p}$ and bias terms $b^{(1)} \in \mathbb{R}^p$ and $b^{(2)} \in \mathbb{R}^n$. The prediction on an input $x \in \mathbb{R}^n$ is

$$h_{W,b} = f(W^{(2)} f(W^{(1)} x + b^{(1)}) + b^{(2)})$$

Given training examples $x^{(1)}, \ldots, x^{(m)} \in \mathbb{R}^n$ the objective function is

$$J(W, b) = \frac{1}{m} \sum_{i=1}^{m} \ell(h_{W,b}(x^{(i)}), x^{(i)}) + \lambda \psi(W, b) + \beta \sum_{j=1}^{p} \phi(\hat{\rho}_j)$$

where

$$\hat{\rho}_j = \frac{1}{m} \sum_{j=1}^{m} f\left( (W_j^{(1)})^T x^{(i)} + b_j^{(1)} \right)$$

is the average activation of the $j$th hidden unit over the training set and $\ell : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a loss function. The function $\psi$ is a regularizer.

# A sample slide

A displayed formula:

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$$

An itemized list:

- itemized item 1
- itemized item 2
- itemized item 3

**Theorem**

*In a right triangle, the square of hypotenuse equals the sum of squares of two other sides.*