A Farewell to Structural Rigidity

Evolving the Starting State of Autoencoders

Christopher Ross   Jonathan Brant   Zak Roessler

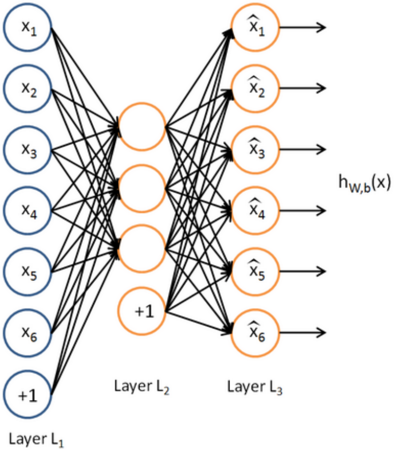
October 28, 2015

Introduction

- Image Processing/Classification
  1. Each pixel is a separate input
  2. ANN inputs grows exponentially as resolution increases
  3. Method of dimensionality reduction needed
- Autoencoders learns compressed representation
- Traditionally, autoencoder hidden layer topology has been hand-crafted
- Our approach evolves autoencoder structure that can then be optimized
  1. Hidden layer of autoencoders evolved (using HyperNEAT)
  2. Backpropagation optimizes evolved autoencoder

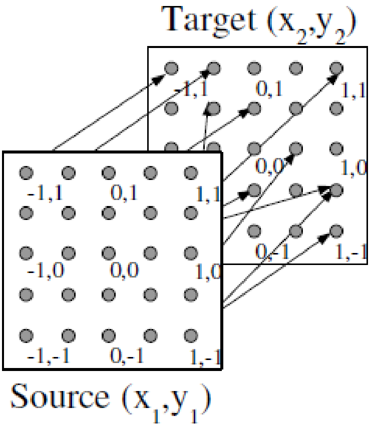
Autoencoders

- Autoencoders attempt to extract most salient features from visible layer
- Typical layers include:
  1. Input (visible) layer where uncompressed features are received
  2. Hidden layer where compressed features are stored
  3. Output layer where reconstruction error is calculated
- Most often trained using backpropagation with the intent of minimizing reconstruction error



HyperNEAT

- Indirect encoding of HyperNEAT through CPPNs substantially reduces search space
  1. We're no longer looking at each pixel individually
- Substrate configuration hand-designed to capture domain geometry
  1. State-space sandwich substrate ideal for visual mapping
- Ability to scale with little-to-no loss of function ideal for image processing at varying resolutions

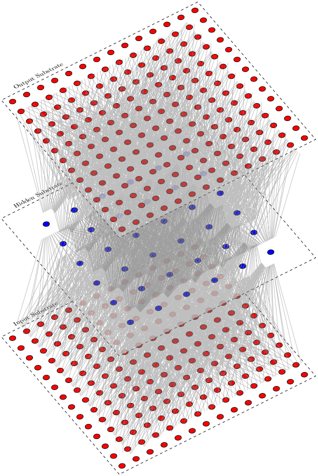


Evolution of Autoencoders

- Two-phase process:
  1. HyperNEAT evolves initial autoencoder
  2. Backpropagation fine tunes autoencoder weights
- Results will be analyzed:
  1. Quantitatively - reduction of reconstruction error
  2. Qualitatively - how closely does reconstructed image resemble the original

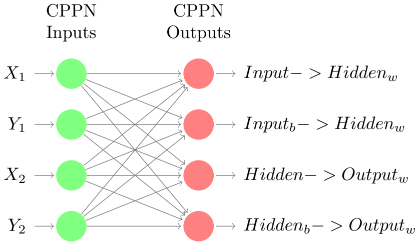
Substrate Configuration

- Three-tiered state-space sandwich substrate
  1. Input sheet - all points in 2-dimensional image (size varies based on image resolution)
  2. Hidden sheet - compressed feature vector
  3. Output sheet - reconstructed image at original resolution



Minimal CPPN

- CPPN inputs are cartesian coordinates between source and target sheet
  - X_1 and Y_1 is point on source sheet
  - X_2 and Y_2 is point on target sheet
- CPPN queries the substrate twice:
  - Once for connection weight and bias weight between input and hidden sheet
  - Once for connection weight and bias weight between hidden and output sheet



Current State

- Standard NEAT initially attempted
  - Slow evolution due to massive search space
  - Qualitatively poor image reconstruction
  - Reconstruction reached 90% accuracy, but only because most pixels were black
- HyperNEAT results?