Designing Neural Networks

Jeff Hajewski advised by Prof. Suely Oliveira
Department of Computer Science
University of Iowa

What are neural networks?

A neural network is a functional approximation to some desired function

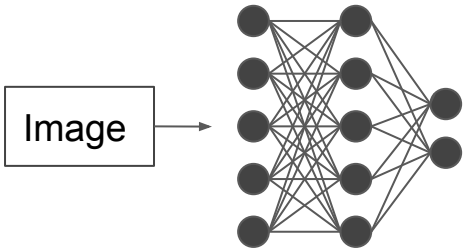
Examples:

- The function that classifies images as cats or dogs
- The function that determines if a human is in a picture
- The function that determines the likelihood of winning a game of chess for a given board state

What are neural networks?

Neural networks are built from layers and activation functions

- Layers are weight vectors where the weights are learned through training
- Activation functions are non-linear functions such as `max(0, x)` or `tanh(x)`



Cat

Dog

$$\arg\max_{i=\text{index 1, index 2}} h_3(h_2(h_1(\omega_1 \cdot x) \cdot \omega_2)) \cdot \omega_3)$$

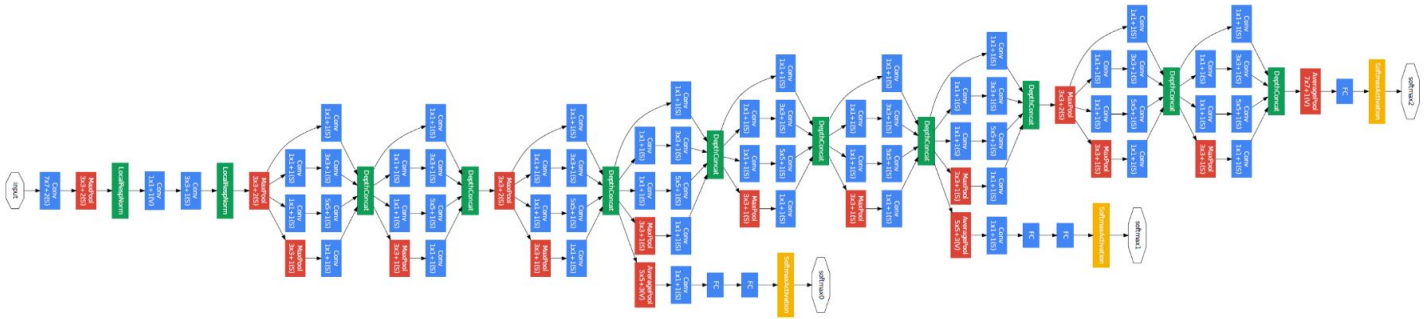
Where are they used?

Neural networks are the driving force behind many applications of artificial intelligence

- Self-driving cars (Tesla, Waymo, Uber)
- Chess, Go, and StarCraft II AIs (see Google DeepMind’s work in these areas)
- Internet Search (ranking algorithm)
- Computer Vision

Neural Network Architecture

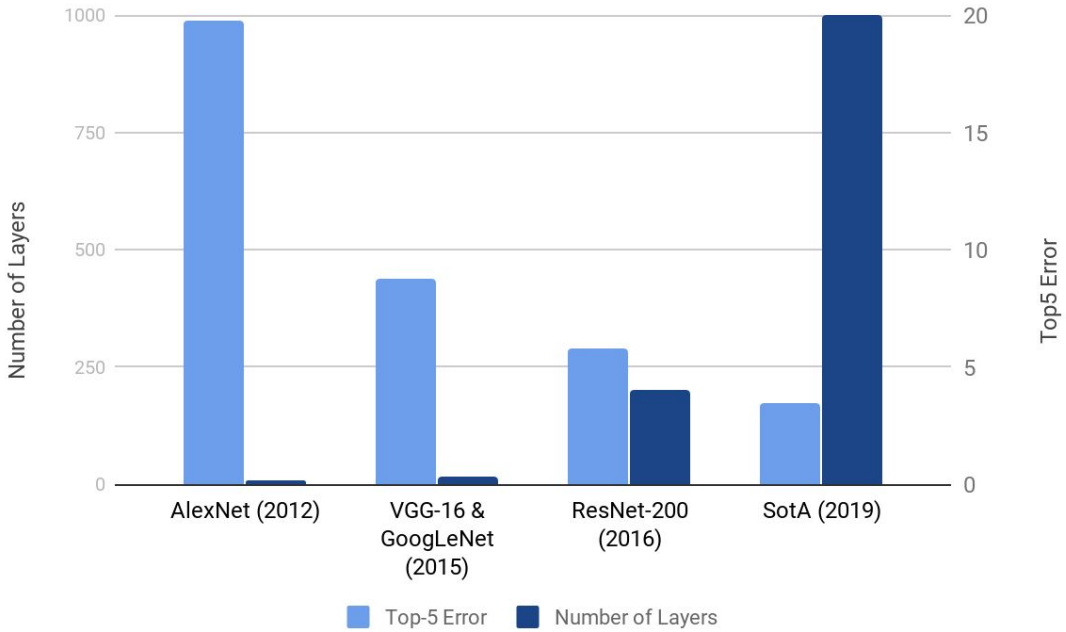
- Layer types: Dense, Convolutional, Pooling, Flatten
- Number of layers
- Size of layers
- Connections between layers (O(n 2^n))
- Activation functions: ReLU, PReLU, leaky-ReLU, tanh, etc.



Going Deeper with Convolutions, Szegedy et al., CVPR 2015

How do we design them?

Guess and check! (seriously!)



Why is designing networks so challenging?

Designing an effective neural network is challenging due to two primary reasons

1. It is difficult to develop an intuition about the interaction between layer types, number of layers, size of layer, activation functions, and connection patterns
2. The candidate neural network must first be trained before it can be evaluated - this training can take anywhere from minutes to hours!

Automating the design of neural networks

The two main approaches to the automated design of neural networks are:

1. Reinforcement learning - train a neural network to learn how to build neural networks
2. Evolutionary algorithms - use ideas inspired by evolution in nature to evolve a population of neural networks over time

Both techniques can achieve state of the art results

This is referred to as **neural architecture search (NAS)**

Evolving neural networks

1. Generate an initial population of neural networks
2. For each network, with some probability either do nothing, mutate the network, or mate the network with another network
3. Train and evaluate each network in the new population
4. Select the best architectures to move on to the next generation
5. Go to step 2

The primary advantage of the evolutionary algorithm is its simplicity and ease of implementation.

Computational requirements

- Suppose the average population size for evaluation is 40 neural networks
- Suppose each neural network takes around 5 minutes to train
- 50 generations of evolution will take nearly **7 days**
- For perspective, many modern works look at *hundreds* of generations
- An 8 GPU machine on Amazon Web Services costs about around $300 per day

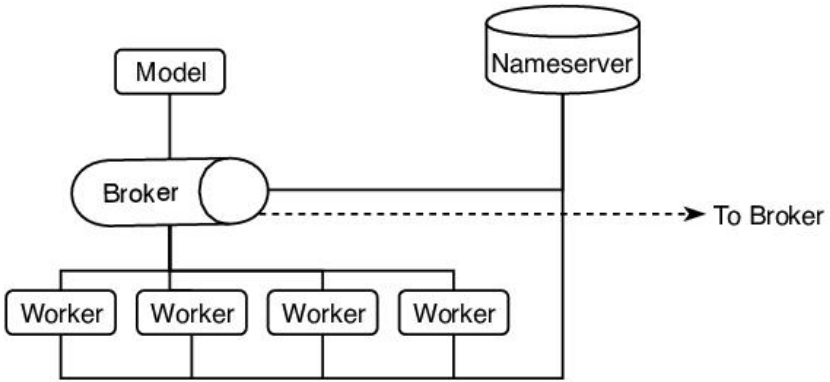
Our work

We take a two-step approach to this problem:

1. Distribute the work across multiple computers
2. Reduce the amount of work being done without sacrificing the effectiveness of the search

Distributed Neural Architecture Search

- Communication via Protocol Buffers and gRPC
- Workers can arbitrarily join and leave the system
- Supports multiple models



Improving efficiency

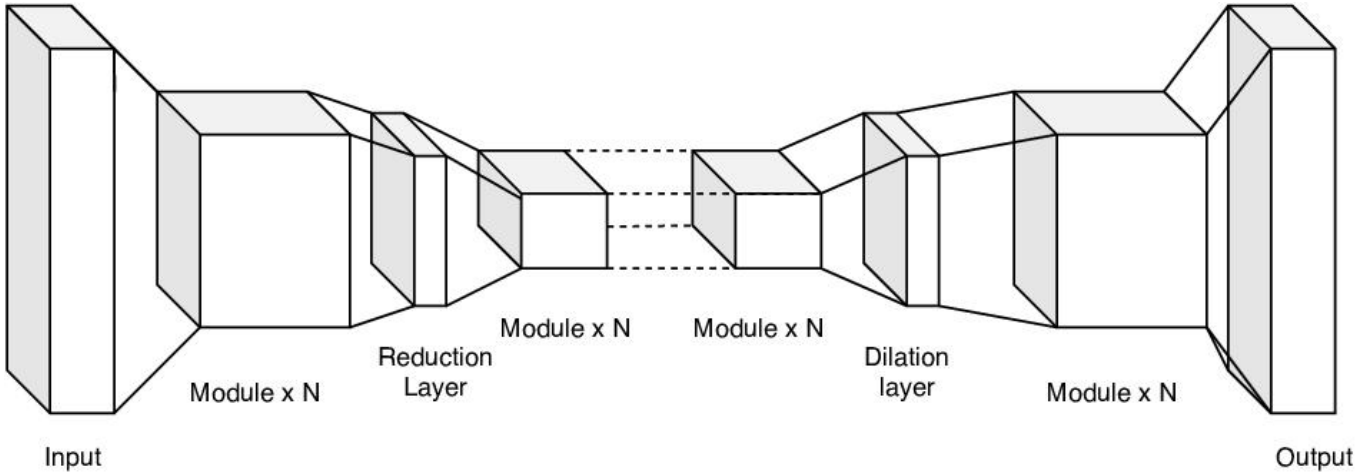
Works

- Making genetic description of neural networks immutable
- Cache prior results

Exploring

- Early stopping
- Reduce the amount of training data

Evolving an autoencoder



Autoencoders are useful for feature selection, noise reduction, and machine translation

Noise reduction

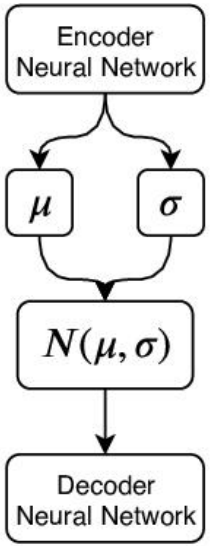
Original

Input

Output

Evolving a variational autoencoder

Variational autoencoder learn a parameterized distribution of the data



Learned distribution

5 epoch search

2 epoch search

Interested in parallel/distributed machine learning?

Research group

- Prof. Suely Oliveira suely-oliveira@uiowa.edu
- Jeff Hajewski (me) jeffrey-hajewski@uiowa.edu
- Cory Kromer Edwards
- Geoff Converse

Feel free to reach out!