Genome - SD1

Sam Henkes, Karthik Subbarao, Brad Akin,
Will Doughty, Jichuan Zhang

Outline

- Problem & Solution
- Our Progress
- Deep Learning & Genetic Algorithm
- Software Tools
- Implemented Algorithm
- Demo
- Results
- Limitations
- Future Works

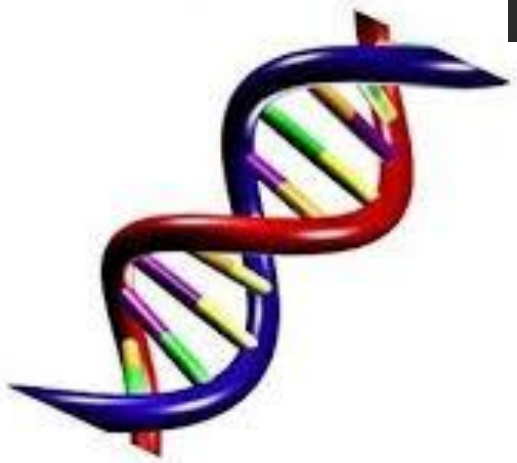
Problem Being Addressed

Finding optimal model parameters for Deep learning Convolutional Neural Networks (CNN) require:

- extensive manual testing
- expensive resources
- 40% of companies take >30 days

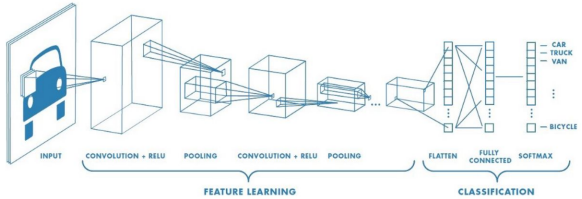
Proposed Solution

Integrate a Genetic Algorithm with Deep Learning to quickly find optimal model parameters for CNNs



Our Progress

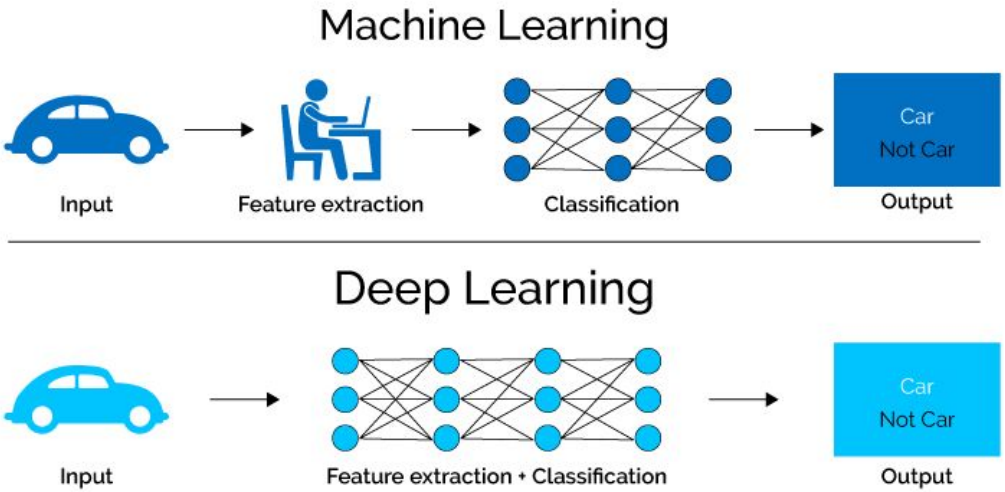
- Week 1 - 2: Genetic Algorithm
- Week 3 - 4: Deep Learning
- Week 5 - 6: Integration of GA with DL
- Week 7 - 9: Increasing complexity of Algorithm
- Week 10 - 11: Testing on Vermeer Data & Improving Algorithm



coursera

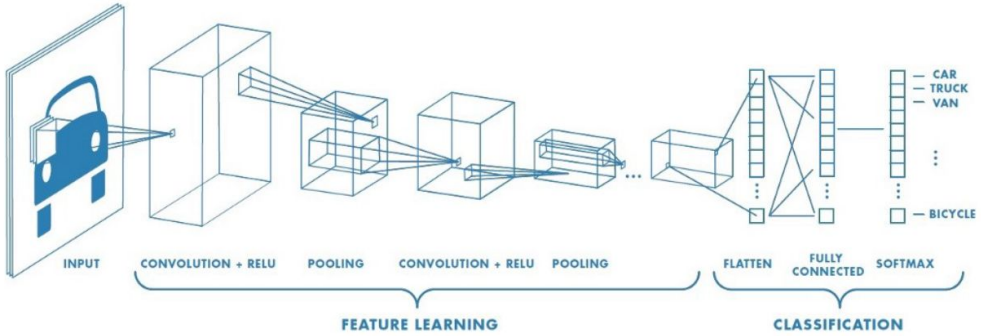
Deep Learning

- Artificial Neural Network
- Levels of neurons
- Performance improves with practice



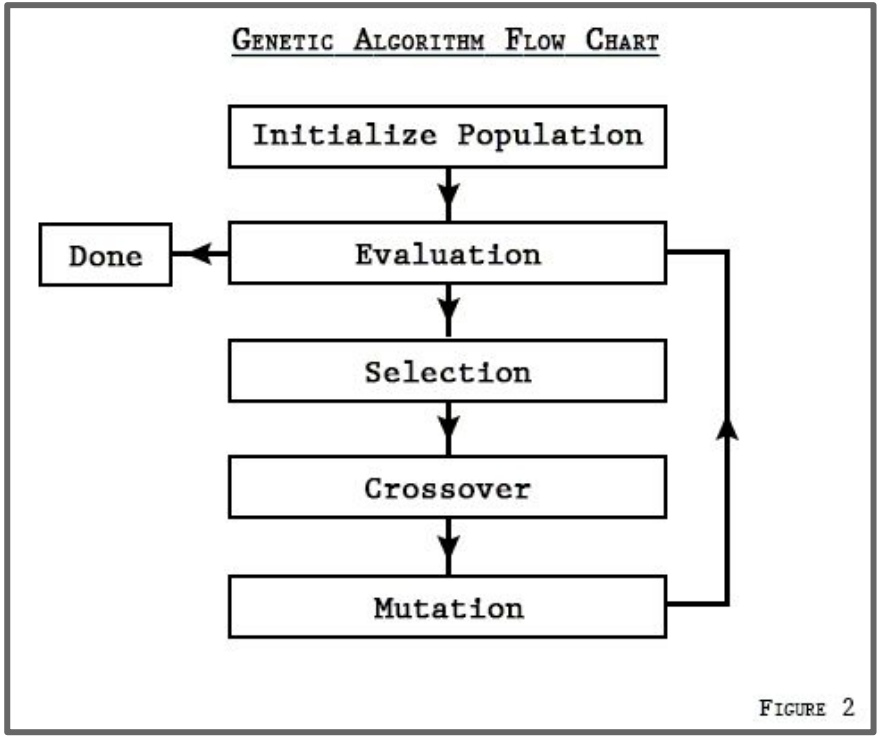
Deep Learning

- CNN
  - Classify Images
  - Built in TensorFlow



Genetic Algorithm

- Parameters
  - Population Size
  - Generation Count
  - Population Retained
- Variables
  - Batch Size
  - Number of Layers, Filters, Nodes
  - Etc.
- Fitness Function



Genetic Algorithm

- Process of Natural selection
- Crossover
- Mutation

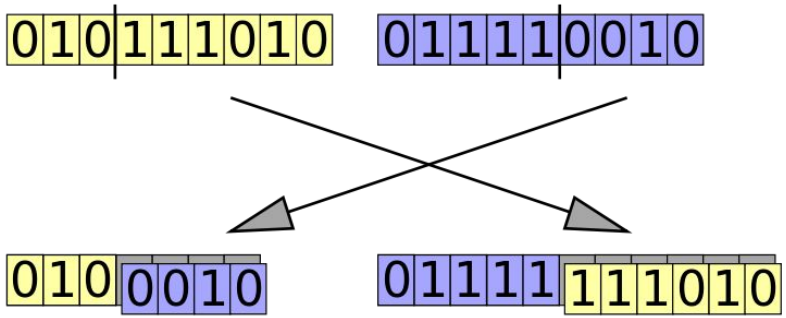
Before Mutation

A5	1	1	1	0	0	0

After Mutation

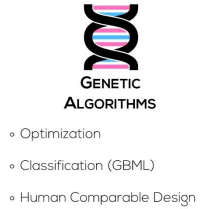
A5	1	1	0	1	1	0

Crossover example



Software / Tools

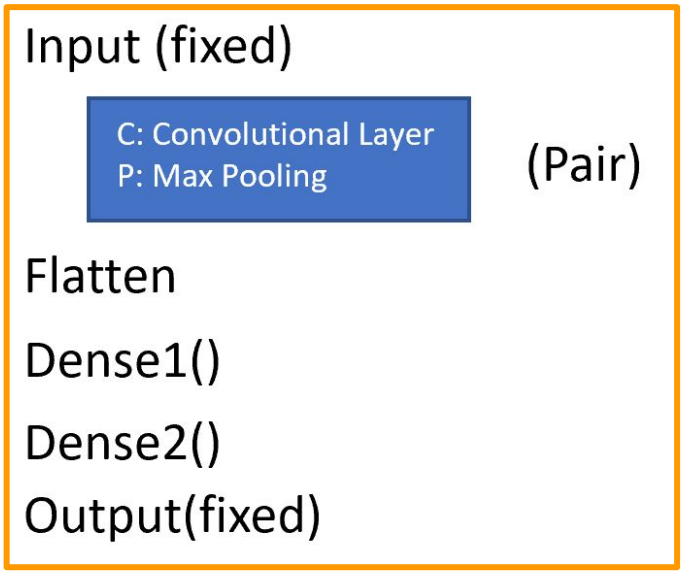
- Genetic Algorithm
- TensorFlow
- Keras
- CNN model architecture
- Python
- Jupyter notebook / IDE
- Colab notebook
- GCP VMs



Implemented Algorithm

- Implement a Genetic Algorithm
- Fitness Function
  - Integrate with Genetic Algorithm
  - Create baseline model
- Add variables for Genetic Algorithm to optimize
  - Batch Size, Activation Function, Optimizer
  - Layers - Filter, Node, Pool Size

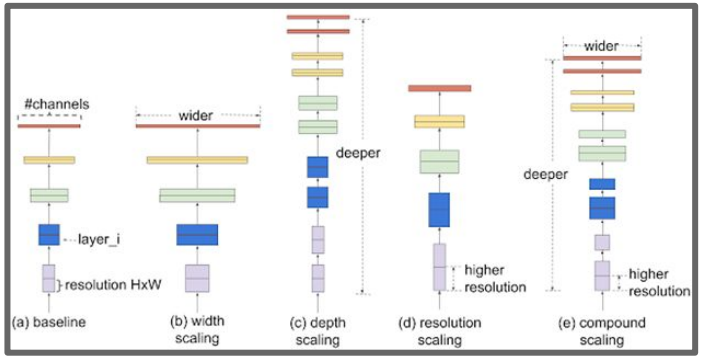
```
BatchList = [32,64,128,256]
Optimizers = ['Adam','Nadam','Adadelta','Adamax']
Activations = ['relu','tanh','selu','elu']
```



Baseline model

Implemented Algorithm

- Variables
  - Layers -
    - Filter size, Number of Nodes, Pooling Size
    - Number of Pairs & Number of layers in each Pair



Input (fixed)

C: Convolutional Layer
P: Max Pooling
(Pair)

Flatten

Dense1()

Dense2()

Output(fixed)

Baseline model

Input (fixed)

C: Convolutional Layer
C: Convolutional Layer
C: Convolutional Layer
P: Max Pooling
(Pair)

C: Convolutional Layer
P: Max Pooling
(Pair)

C: Convolutional Layer
C: Convolutional Layer
P: Max Pooling
(Pair)

.
.
.

Flatten

Dense1()

Dense2()

Output(fixed)

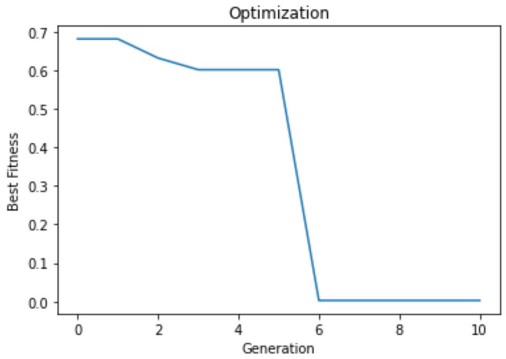
Generated model

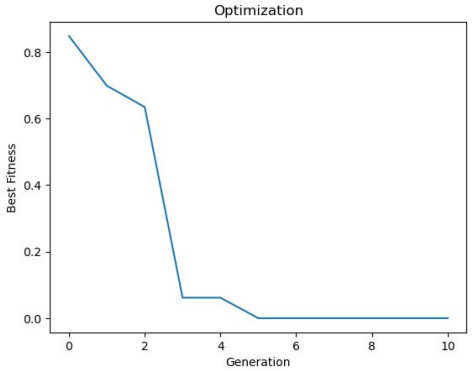
Demonstration

- Vermeer Data
- Algorithm
- GCP
- Results

Results

- Genetic Algorithm by nature reaches optimal solution:
  - Validation Loss gradually decreases
- Best Model & Parameters are saved
- Successfully works on Vermeer Data

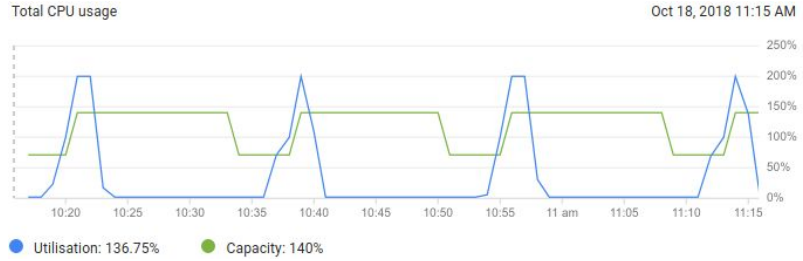


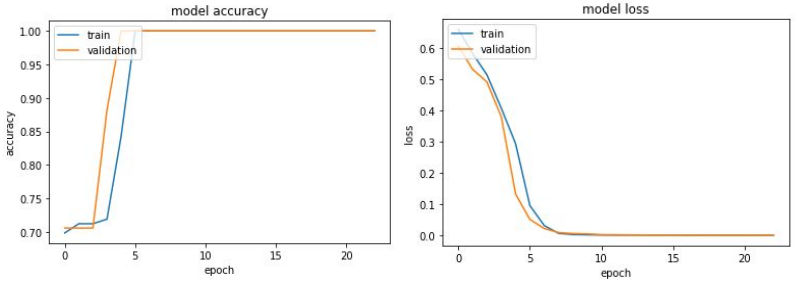


Limitations

- Uses lots of memory
- Needs lots of compute
- More variables to add
  - Learning Rate, Dropout Layer, Patience for Early Stopping
- Tested algorithm on a few datasets for convergence

```
ResourceExhaustedError:  OOM when allocating tensor with shape
```

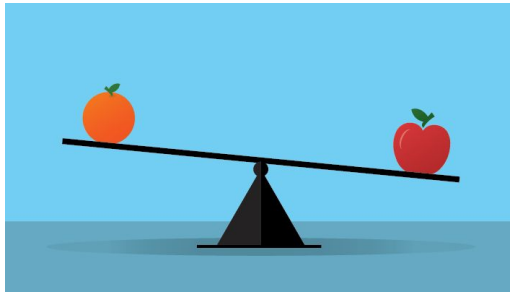




Future Works

- Improvement:
  - Add more hyper parameter variables
  - Improve algorithm for better convergence

- Testing:
  - Test on new Datasets
  - Run with different population size, generations



- Compare results with other Research Algorithms
- Write a paper on the Algorithm