Evolutionary Neural Network for data classification

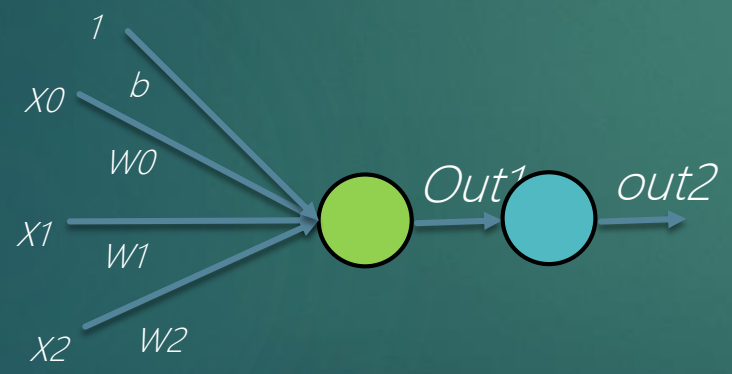
MAJID NASIRI MANJILI & AHMAD MOHAMMADI
PROFESSOR PAZOKI
SHAHID RAJAEE TEACHER TRAINING UNIVERSITY

Ev and NN

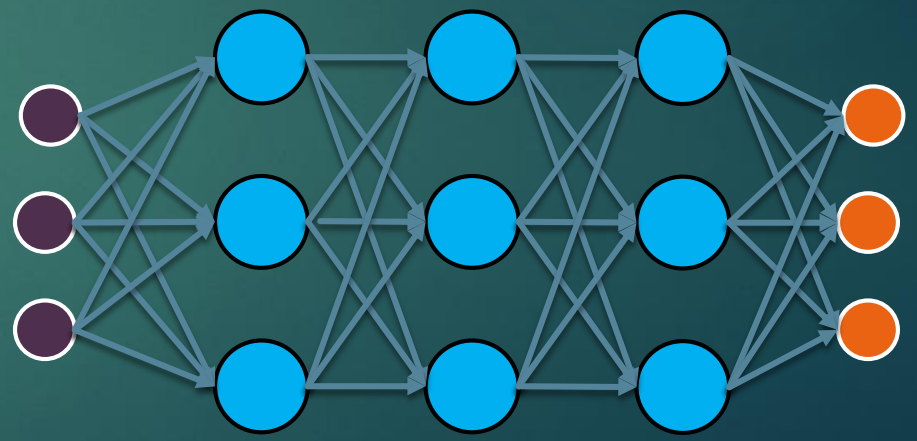
- Evolutionary algorithm and neural networks are both inspired by computation in biological system
- Neural networks and genetic algorithms are two techniques for optimization and learning
- We use Evolutionary algorithm to configure the neural networks

Neural networks

- A computational model consisting of a number of connected elements, known as neurons
- Have input, hidden and output layers that contain many neuron in each layer
- Each of the inputs is modified by a value associated with the connection called weight
- A neuron apply activation function to input and produce output

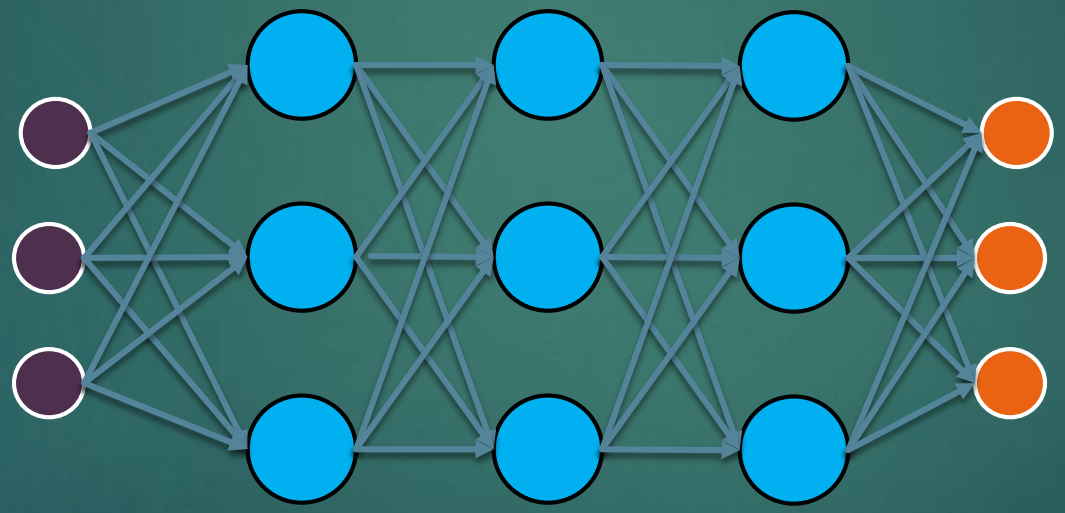


*Out1 = X0*W0 + X1 * W1 + X2* W2 + b*
Out2 = f(out1)



EV usage in NN

- Learning and compute the weights
- Find the optimal number of layer and neurons per layer
- Tune the parameters like learning rate and activation function



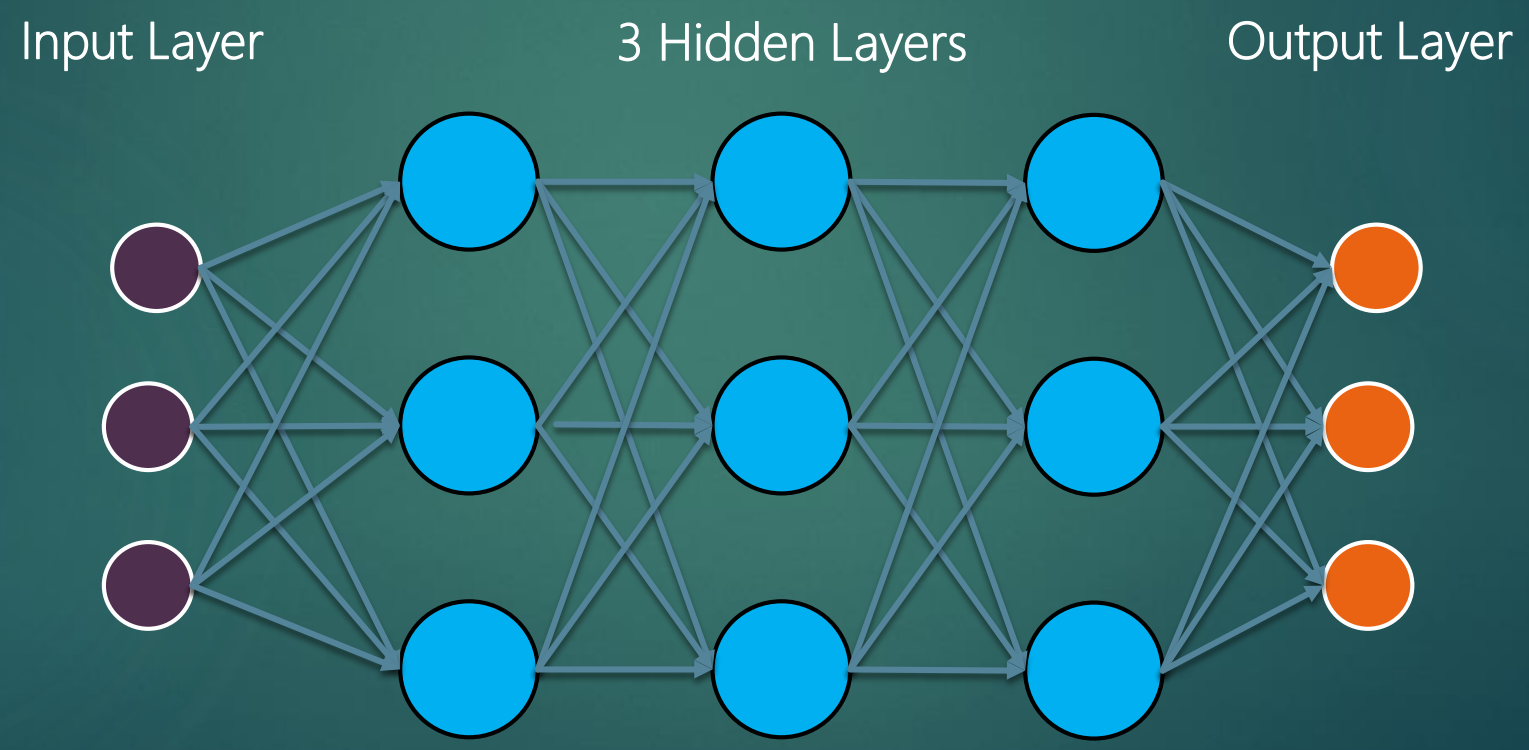
Example

we have four parameters with five possible settings each and it takes five minutes to train and evaluate

- To try them all would take (5**4) * 5 minutes, or 3,125 minutes, or about 52 hours
- Consider we use a genetic algorithm to evolve 10 generations with a population of 20 with a plan to keep the top 25%, so ~8 per generation. in our first generation we score 20 networks (20 * 5 = 100 minutes). Every generation after that only requires around 12 runs, That's 100 + (9 generations * 5 minutes * 12 networks) = 640 minutes, or 11 hours.

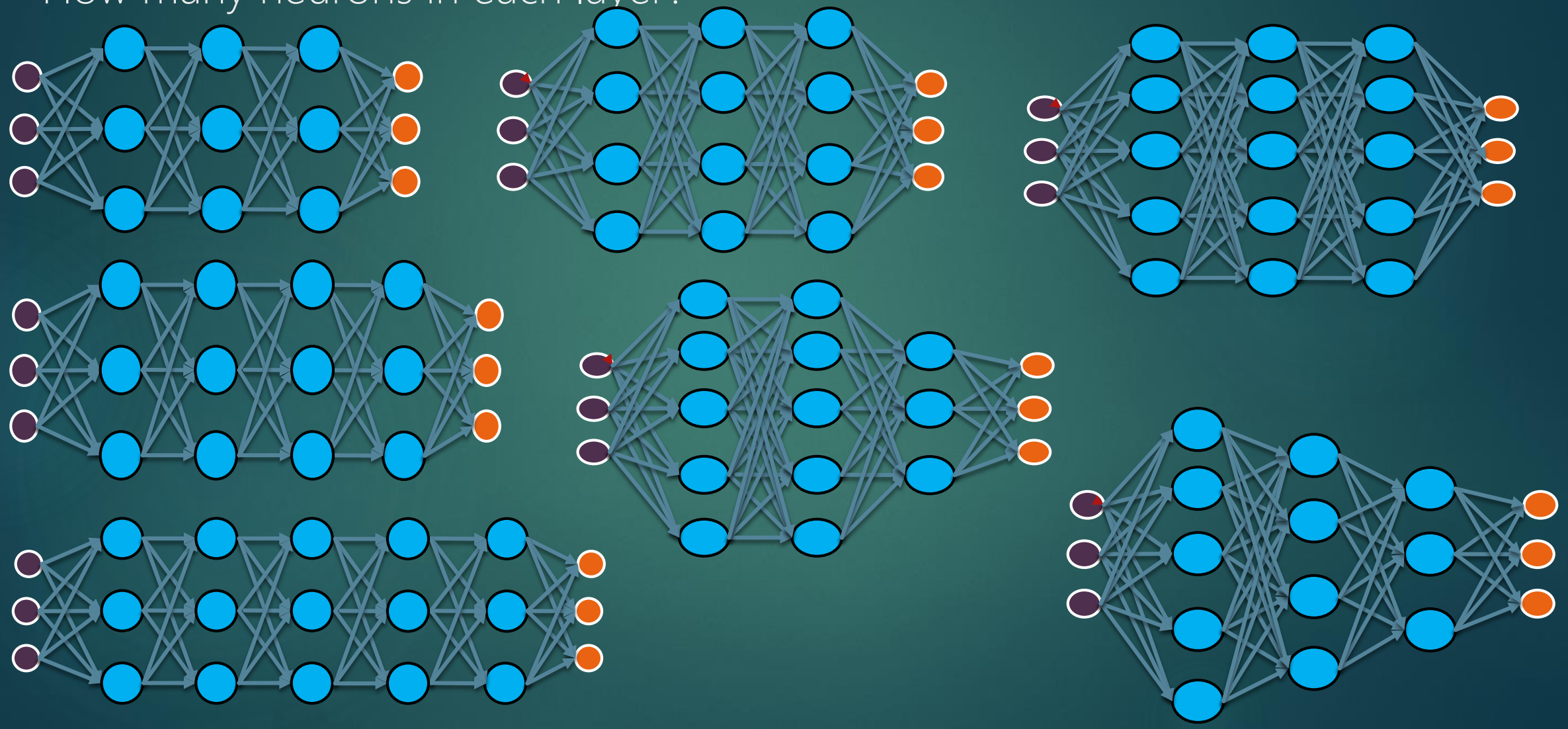
Problem

How many neurons in each layer?



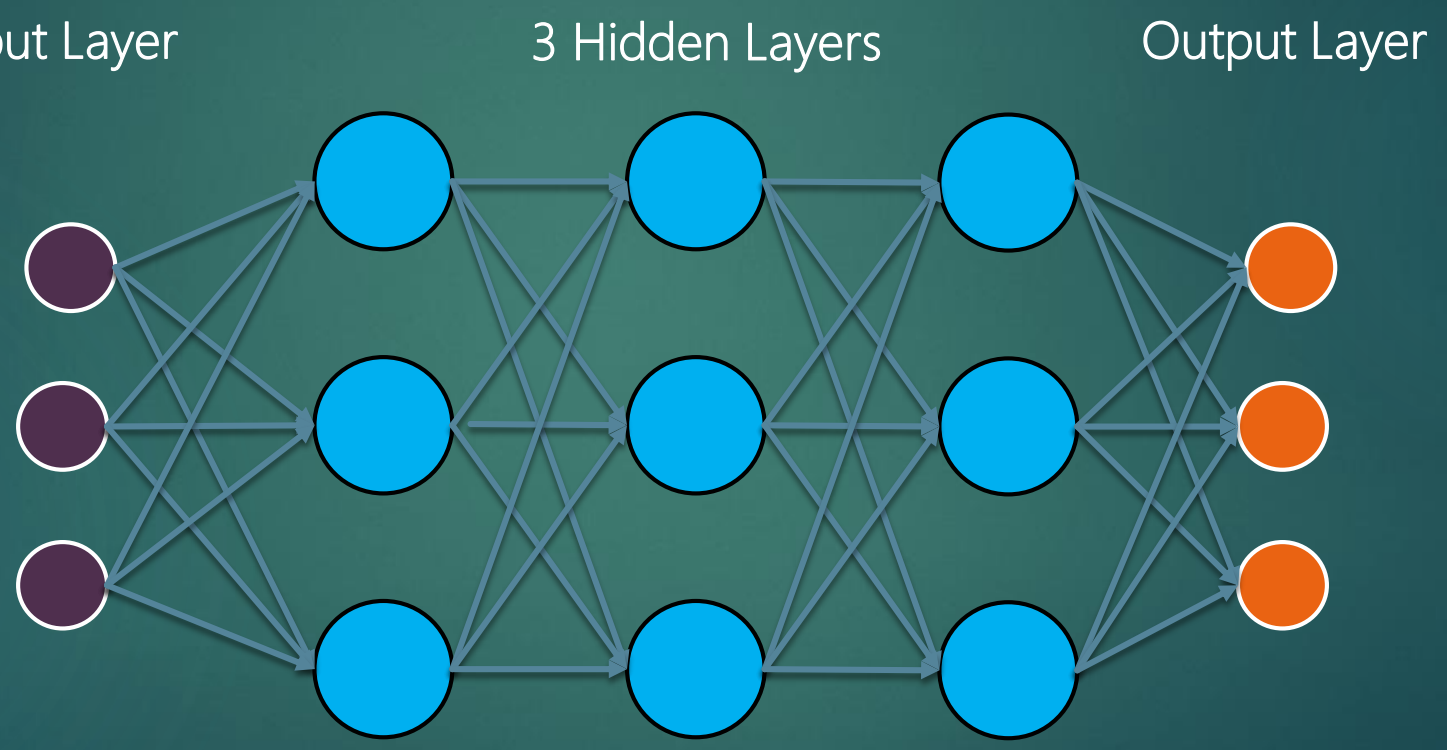
Problem

How many neurons in each layer?



Without Genetic Algorithm

Assume neurons for each layer is [30 ~ 80]

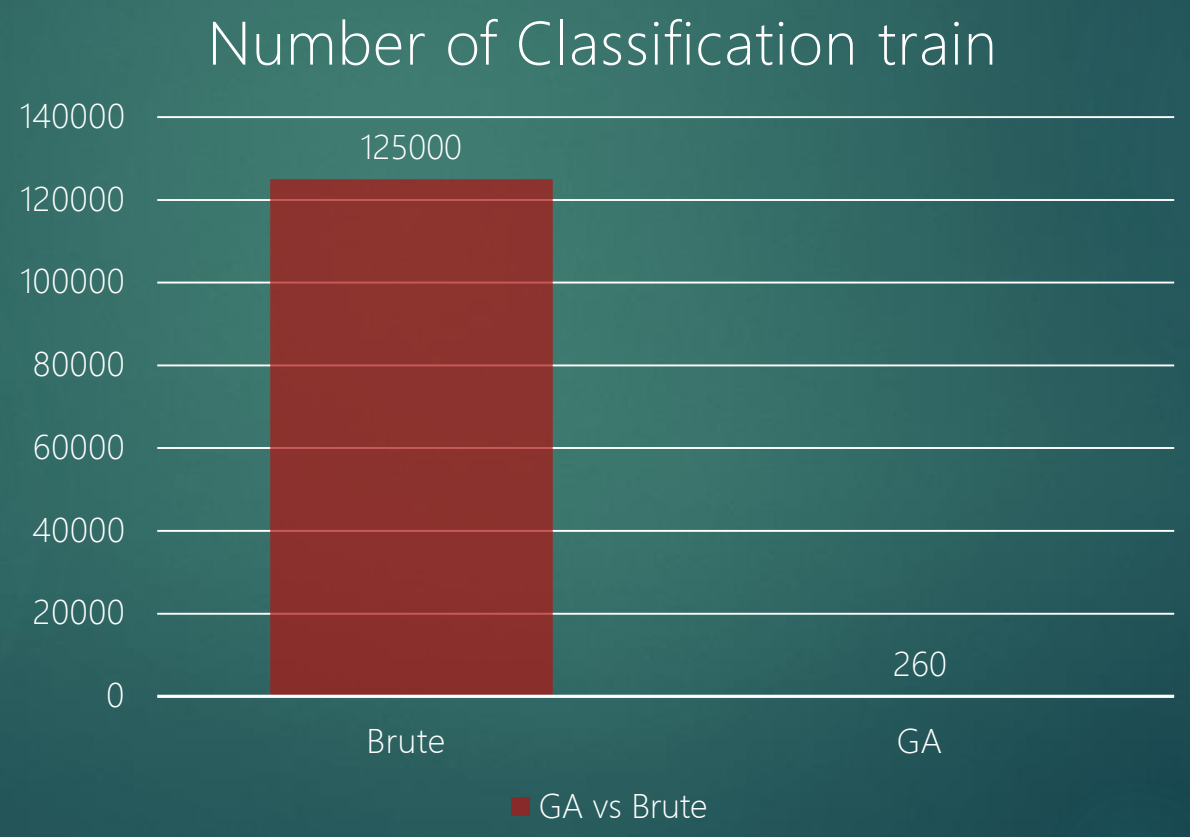


50 * 50 * 50 = 125000

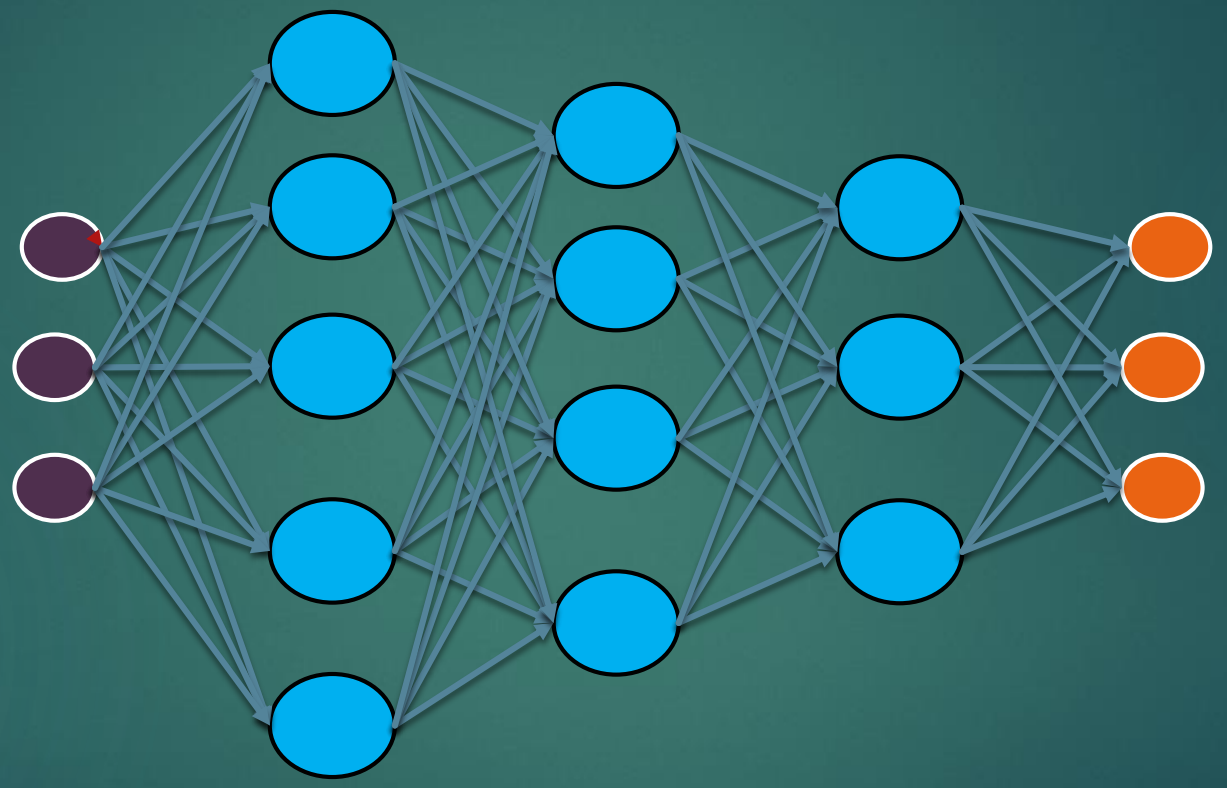
Genetic Algorithms Approach


```
generation_num = 20
20 + generation_num * 12 = 260
```

```
125000/260 = 480
```



Genetic Algorithms Representation



genotype	5	4	3

Parent Selection

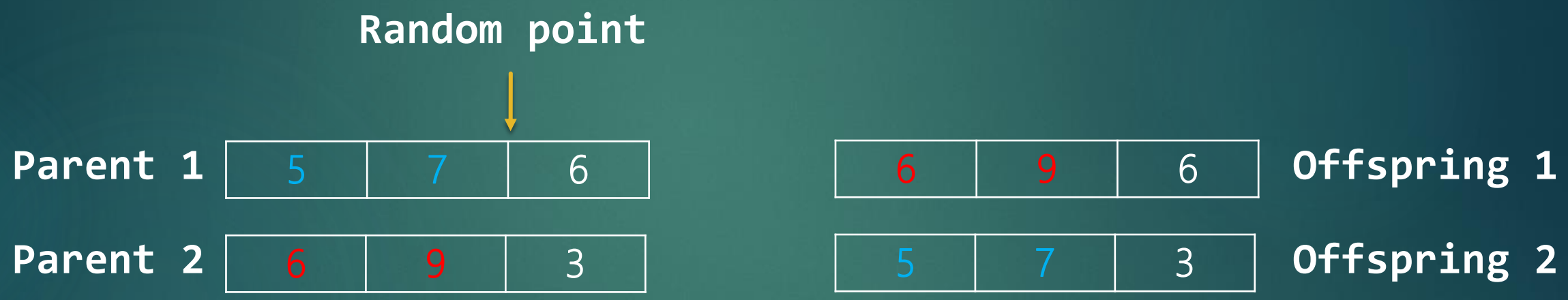
rank	6	4	6
1	10	12	9
2	12	8	6
3	16	12	8
4	15	11	18
5	9	10	3
6	4	7	6
7	20	7	3
8	11	6	7
-	-	-	-
-	-	-	-
19	10	9	9
20	4	3	3

Top five genotype

three Random from losers

Mating Pool

1-Point Crossover



Mutation

Survivor Selection

Eight genotype from Current generation
(Top five genotype + three Random from losers)

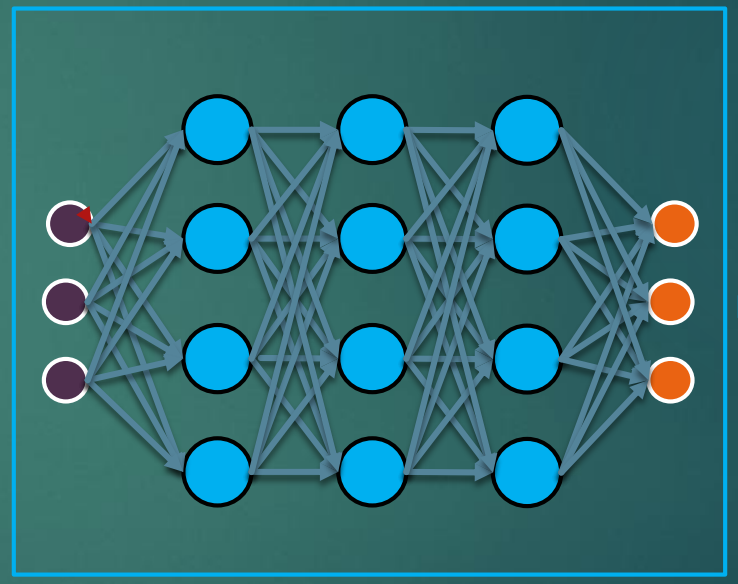
Twelve New offsprings

6	4	6
10	12	9
12	8	6
-	-	-
15	11	18
9	10	3
4	7	6
20	7	3
11	6	7
-	-	-
-	-	-
10	9	9
4	3	3

Genetic Algorithms info

```
Representation ---------------- integer
Recombination ----------------- 1-point crossover
Recombination Probability ---- 90%
Mutation ---------------------- a random new gene
Mutation Probability ---------- 50%
Parent Selection -------------- 5 best and 3 random worst
Survivor Selection ------------ (mu + lambda)
Population Size --------------- 20
Number of Offspring ----------- 20
Initialization ---------------- Random
Termination Condition -------- Affer 20 generation
```

Dataset mnist



0
1
2
3
4
5
6
7
8
9

Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

/home

Editor

mnist_perceptron_net.py | nnec_2.py

```
        #REPEAT UNTIL ( TERMINATION CONDITION is satisfied
        while(self.termination == 0):
            print('----------------------- : generation = ', self.gen)

            #SELECT parents
            self.parent_selection()

            #RECOMBINE pairs of parents
            self.crossover()

            #MUTATE the resulting offspring
            self.mutation()

            #EVALUATE new candidates
            self.offspring_fitness = self.fitness(self.offspring)

            # SELECT individuals for the next generation
            self.survivor_selection()

            self.gen +=1
            if (self.gen == self.max_generation):
                self.termination = 1


st = time.time()
model = evolutionary_computing()
model.run()
print(model.best_fitness_array)
print(model.best_chromosome_array)
et = time.time()
print ('\n%2.2f sec' %  (et-st))
```

Codes in Python
Using Tensorflow
framework

IPython console

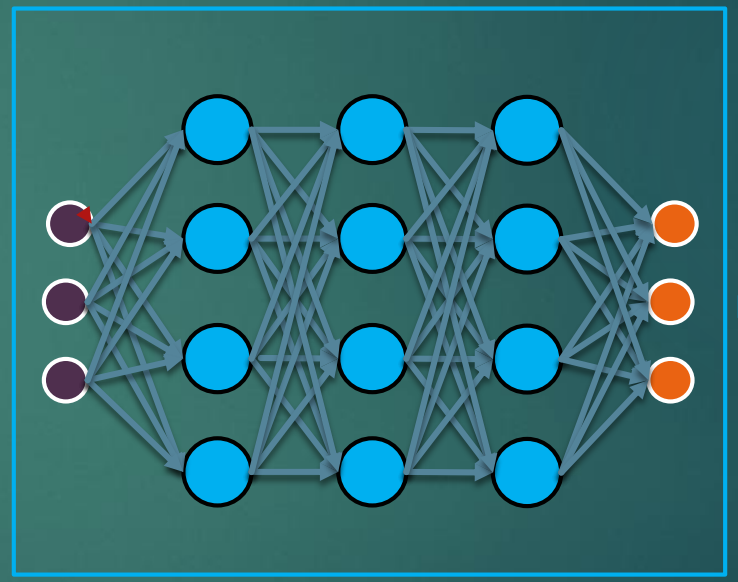
Console 1/A

```
Epoch 0 completed out of 10 loss: 157258.57037
Epoch 1 completed out of 10 loss: 36509.42243
Epoch 2 completed out of 10 loss: 22305.1711979
Epoch 3 completed out of 10 loss: 15648.7125187
Epoch 4 completed out of 10 loss: 11723.3217993
Epoch 5 completed out of 10 loss: 9098.75392985
Epoch 6 completed out of 10 loss: 7206.11012113
Epoch 7 completed out of 10 loss: 5828.9277488
Epoch 8 completed out of 10 loss: 4782.48464894
Epoch 9 completed out of 10 loss: 3986.59642565
chromosomes 11/12 of generation 2 is evaluationg ...
Epoch 0 completed out of 10 loss: 199763.334213
Epoch 1 completed out of 10 loss: 37863.6828995
Epoch 2 completed out of 10 loss: 22484.5542145
Epoch 3 completed out of 10 loss: 15706.565887
Epoch 4 completed out of 10 loss: 11754.6549194
Epoch 5 completed out of 10 loss: 9168.12869143
Epoch 6 completed out of 10 loss: 7341.73067367
Epoch 7 completed out of 10 loss: 5991.75383162
Epoch 8 completed out of 10 loss: 4922.77067721
Epoch 9 completed out of 10 loss: 4086.52029143
[ 0.83579999  0.88380003  0.84100002 ...,  0.88590002  0.88230002
  0.87599999]
[0.87309998273849487, 0.88400000333786011, 0.89020001888275146]
[array([65, 76, 55]), array([75, 69, 54]), array([65, 69, 54])]

3833.49 sec

In [164]: model.best_fitness
Out[164]: 0.89020001888275146

In [165]:
```

Python console | History log | IPython console

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 226 Column: 1 Memory: 82 % CPU: 1 %

Dataset Iris



versicolor

Setosa

Virginica

Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

/home

Editor

iris_nnec_1.py | iris_perceptron_net.py

```python
        #REPEAT UNTIL ( TERMINATION CONDITION is satisfied
        while(self.termination == 0):
            print('---------------------- : generation = ', self.gen)

            #SELECT parents
            self.parent_selection()

            #RECOMBINE pairs of parents
            self.crossover()

            #MUTATE the resulting offspring
            self.mutation()

            #EVALUATE new candidates
            self.offspring_fitness = self.fitness(self.offspring)

            # SELECT individuals for the next generation
            self.survivor_selection()

            self.gen +=1
            if (self.gen == self.max_generation):
                self.termination = 1


st = time.time()
model = evolutionary_computing()
model.run()
print(model.best_fitness_array)
print(model.best_chromosome_array)
et = time.time()
print ('\n%2.2f sec' %  (et-st))
```

IPython console

Console 1/A

```
Accuracy =  0.657143
chromosomes 2/12 of generation 2 is evaluationg ...
Accuracy =  0.357143
chromosomes 3/12 of generation 2 is evaluationg ...
Accuracy =  0.414286
chromosomes 4/12 of generation 2 is evaluationg ...
Accuracy =  0.628571
chromosomes 5/12 of generation 2 is evaluationg ...
Accuracy =  0.357143
chromosomes 6/12 of generation 2 is evaluationg ...
Accuracy =  0.742857
chromosomes 7/12 of generation 2 is evaluationg ...
Accuracy =  0.342857
chromosomes 8/12 of generation 2 is evaluationg ...
Accuracy =  0.657143
chromosomes 9/12 of generation 2 is evaluationg ...
Accuracy =  0.214286
chromosomes 10/12 of generation 2 is evaluationg ...
Accuracy =  0.3
chromosomes 11/12 of generation 2 is evaluationg ...
Accuracy =  0.657143
[ 0.30000001  0.65714288  0.35714287 ...,  0.21428572  0.30000001
  0.65714288]
[0.69999998807907104, 0.69999998807907104, 0.74285715818405151]
[array([17,  9,  6]), array([17,  9,  6]), array([15,  4, 15])]

1318.39 sec

In [43]: model.best_fitness_array
Out[43]: [0.69999998807907104, 0.69999998807907104, 0.74285715818405151]

In [44]: model.best_fitness
Out[44]: 0.74285715818405151

In [45]:
```

Python console | History log | IPython console

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 229 Column: 1 Memory: 45 % CPU: 1 %

Questions ?