

Effects of the Number of Hyperparameters on the Performance of GA-CNN

Tirana Noor Fatyanosa
Masayoshi Aritsugi

Kumamoto University, Japan

Outline

Topics for discussion

Introduction

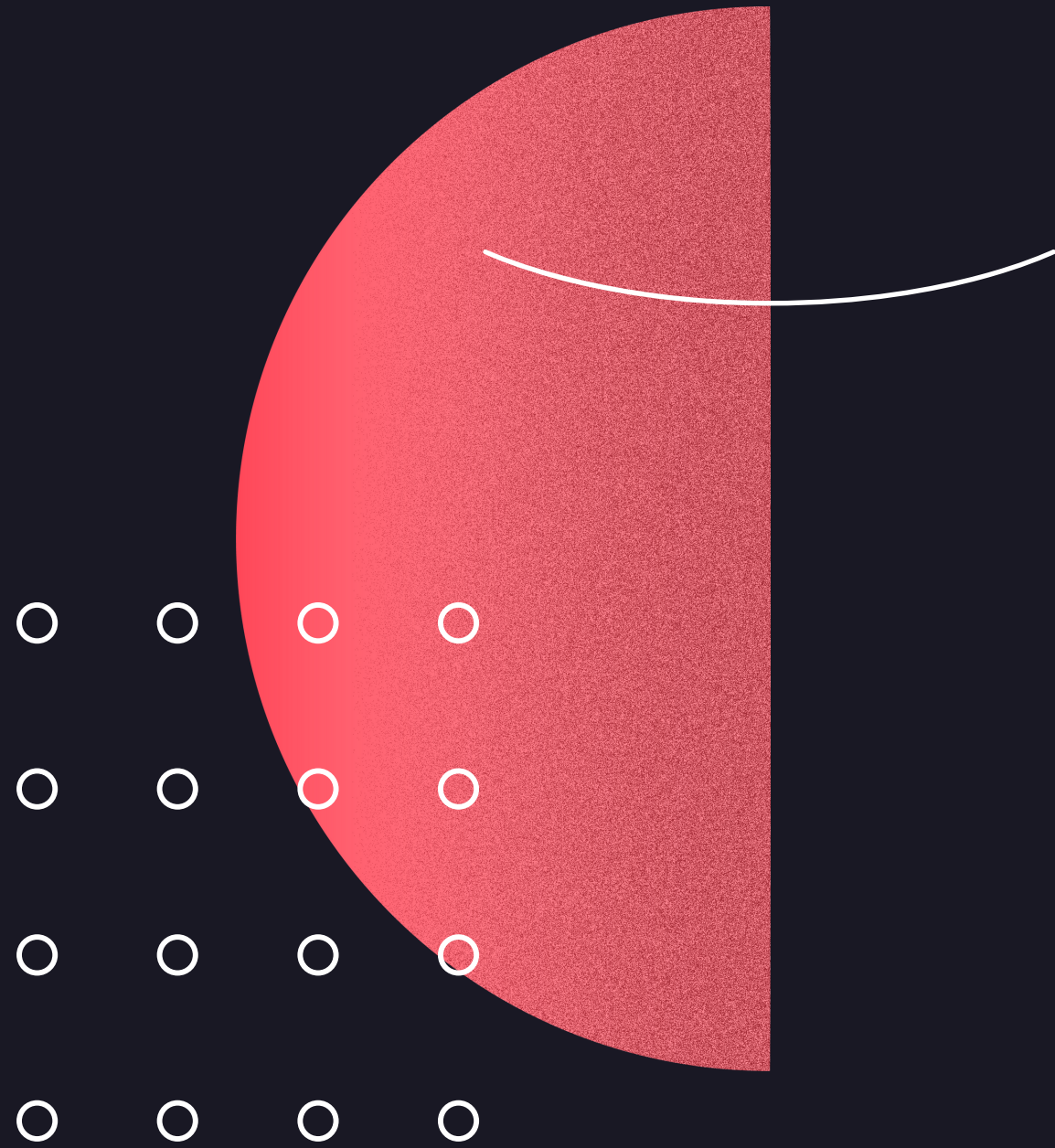
Related Work

CNN Hyperparameters and Architecture

GA-CNN

Results and Discussion

Conclusions and Future Work



INTRODUCTION





HYPERPARAMETERS



ML MODEL



SCORE



n_epochs = 4
batch_size = 64
optimizer = 'sgd'



80%



n_epochs = 10
batch_size = 250
optimizer = 'adam'



90%



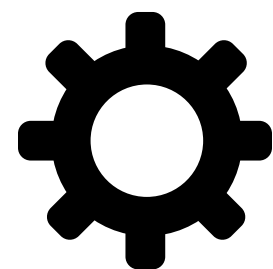
HYPERPARAMETERS



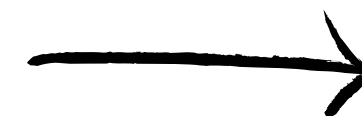
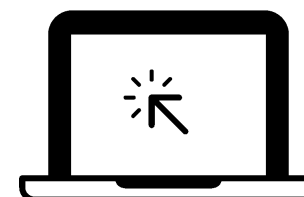
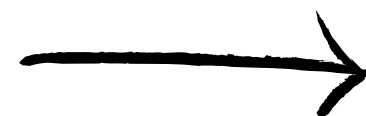
ML MODEL



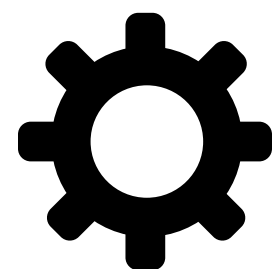
SCORE



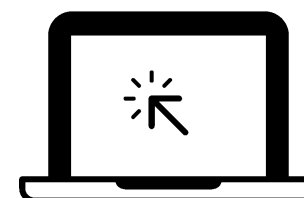
n_epochs = 4
batch_size = 64
optimizer = 'sgd'



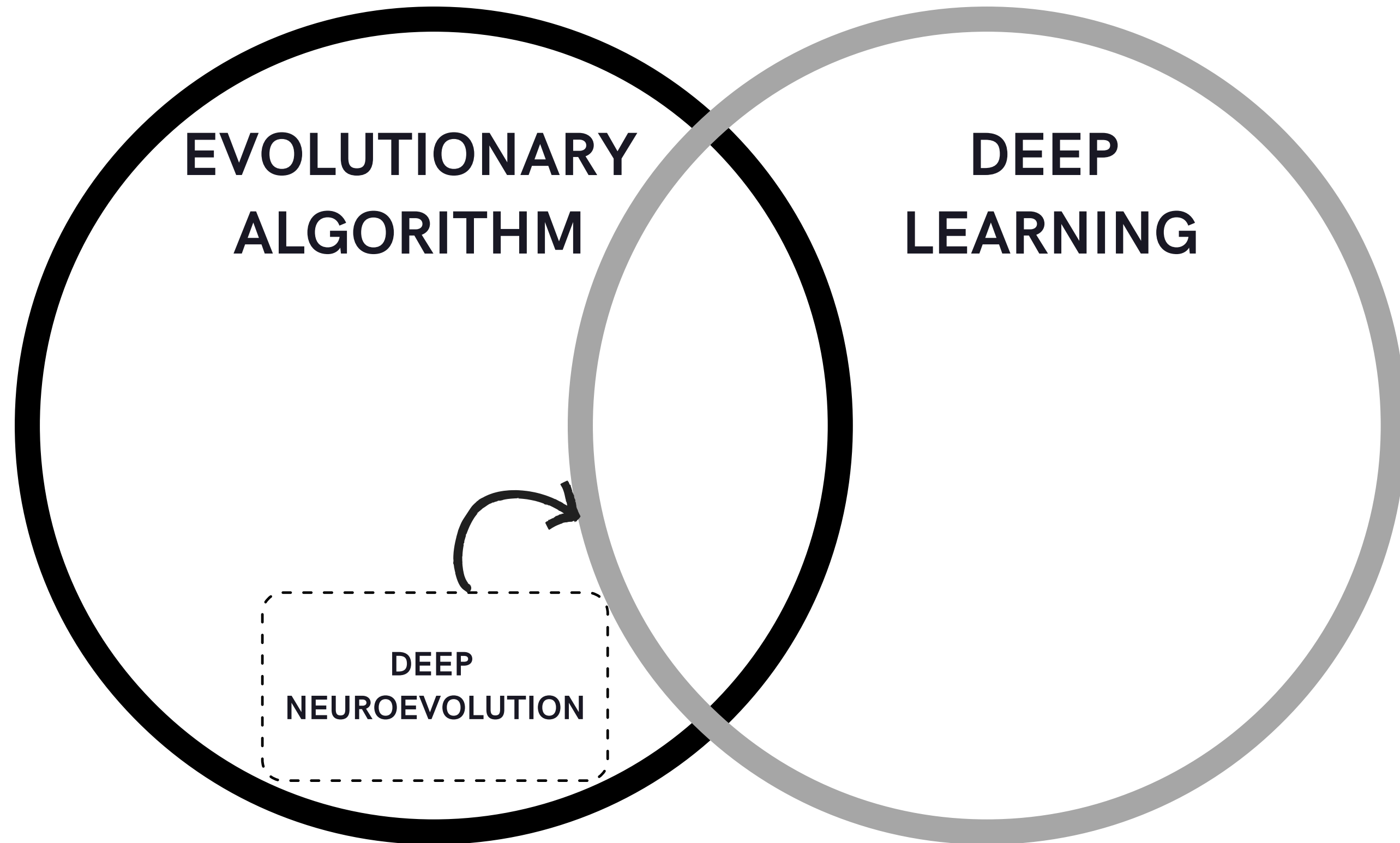
80%

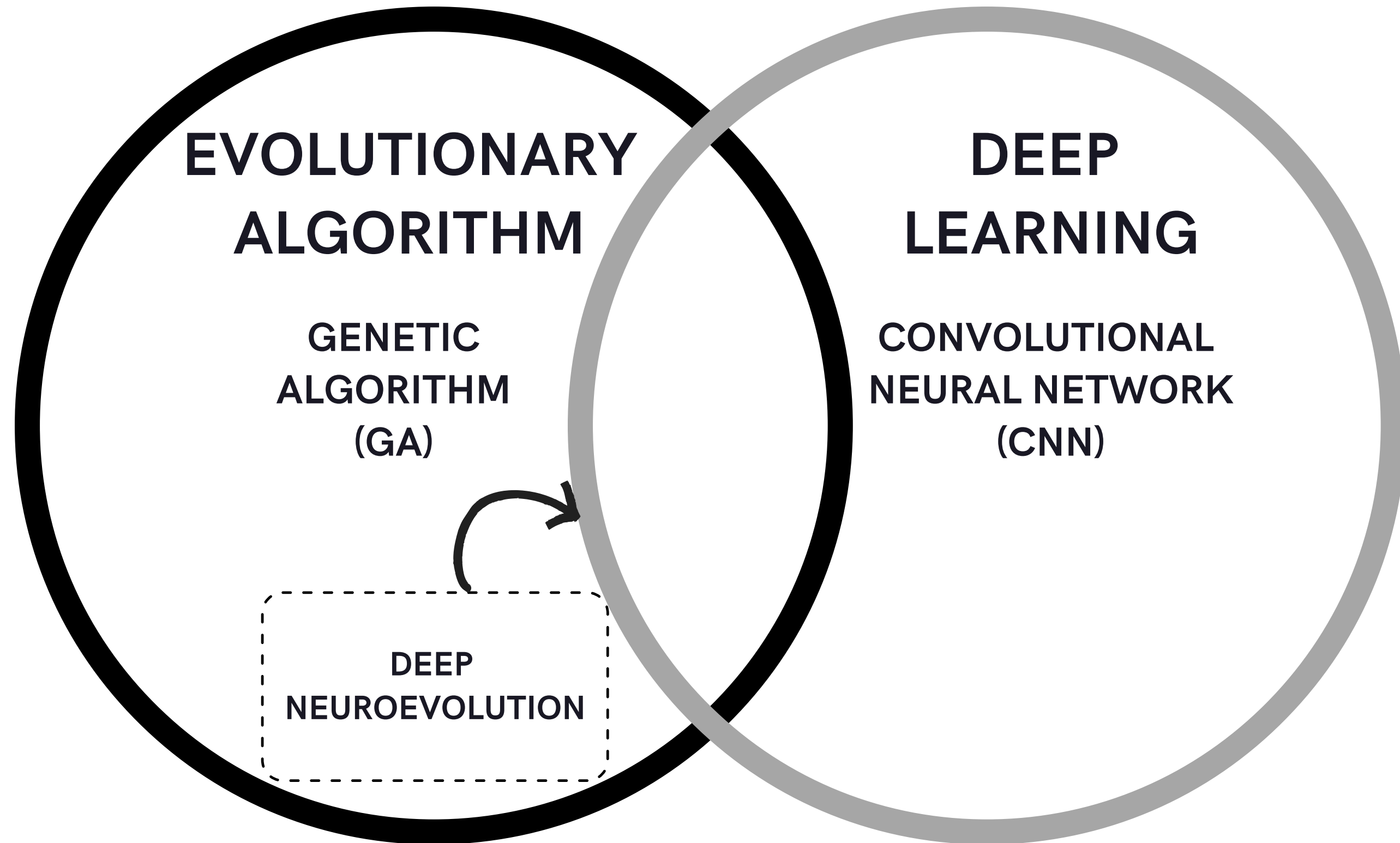


n_epochs = 10
batch_size = 250
optimizer = 'adam'



90%



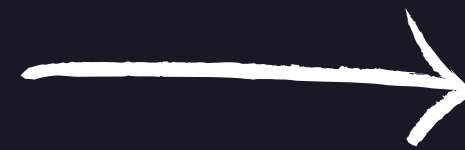


HYPERPARAMETERS

n_epochs
learning_rate
batch_size
optimizer
n_conv_layer
n_dense_layer



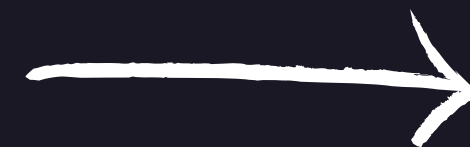
GA-CNN



SCORE



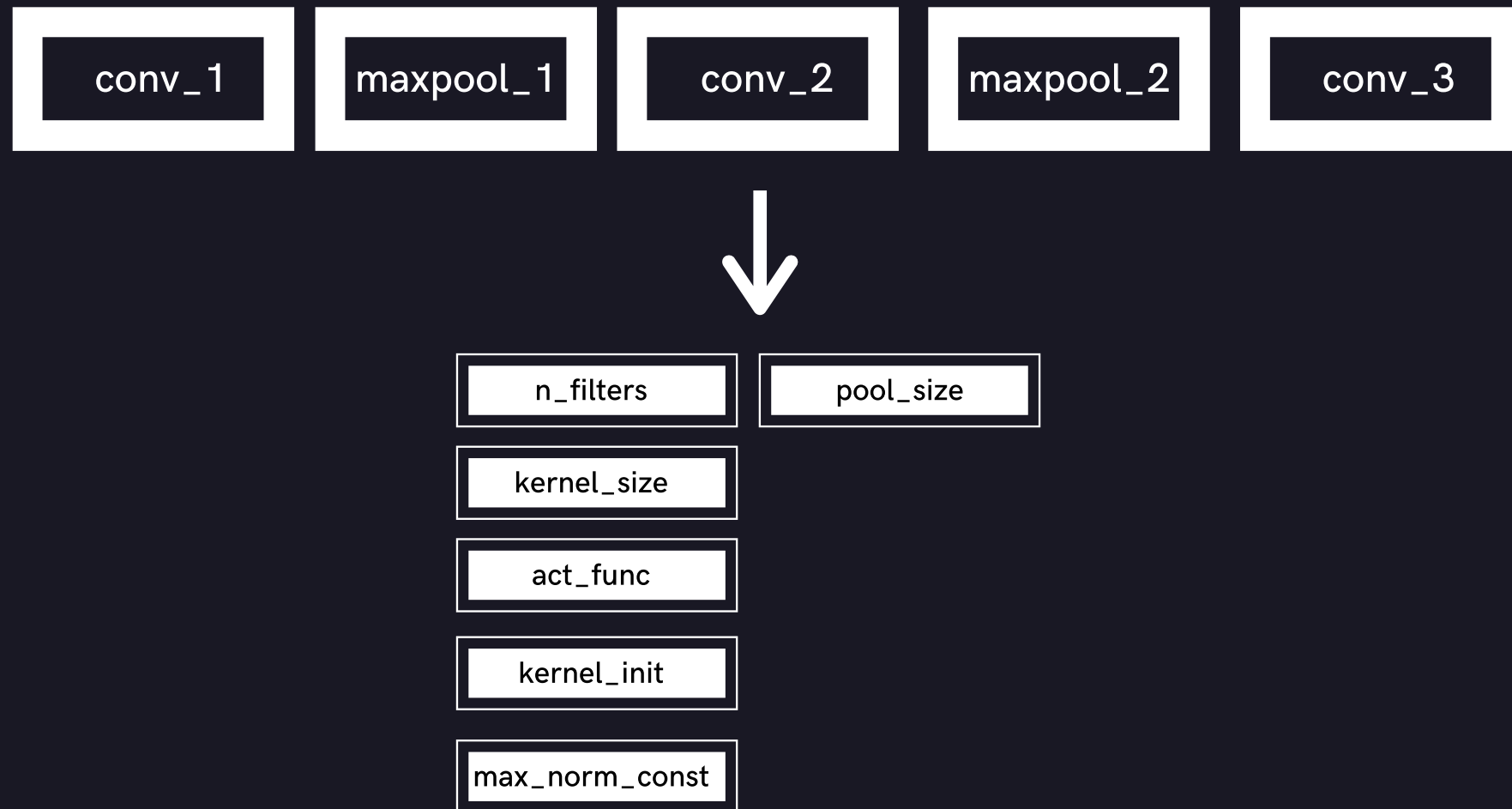
n_epochs
learning_rate
batch_size
n_filters
kernel_size
momentum
optimizer
n_conv_layer
act_func_conv
n_dense_layer
n_neurons



SAME HYPERPARAMETERS FOR EACH LAYER

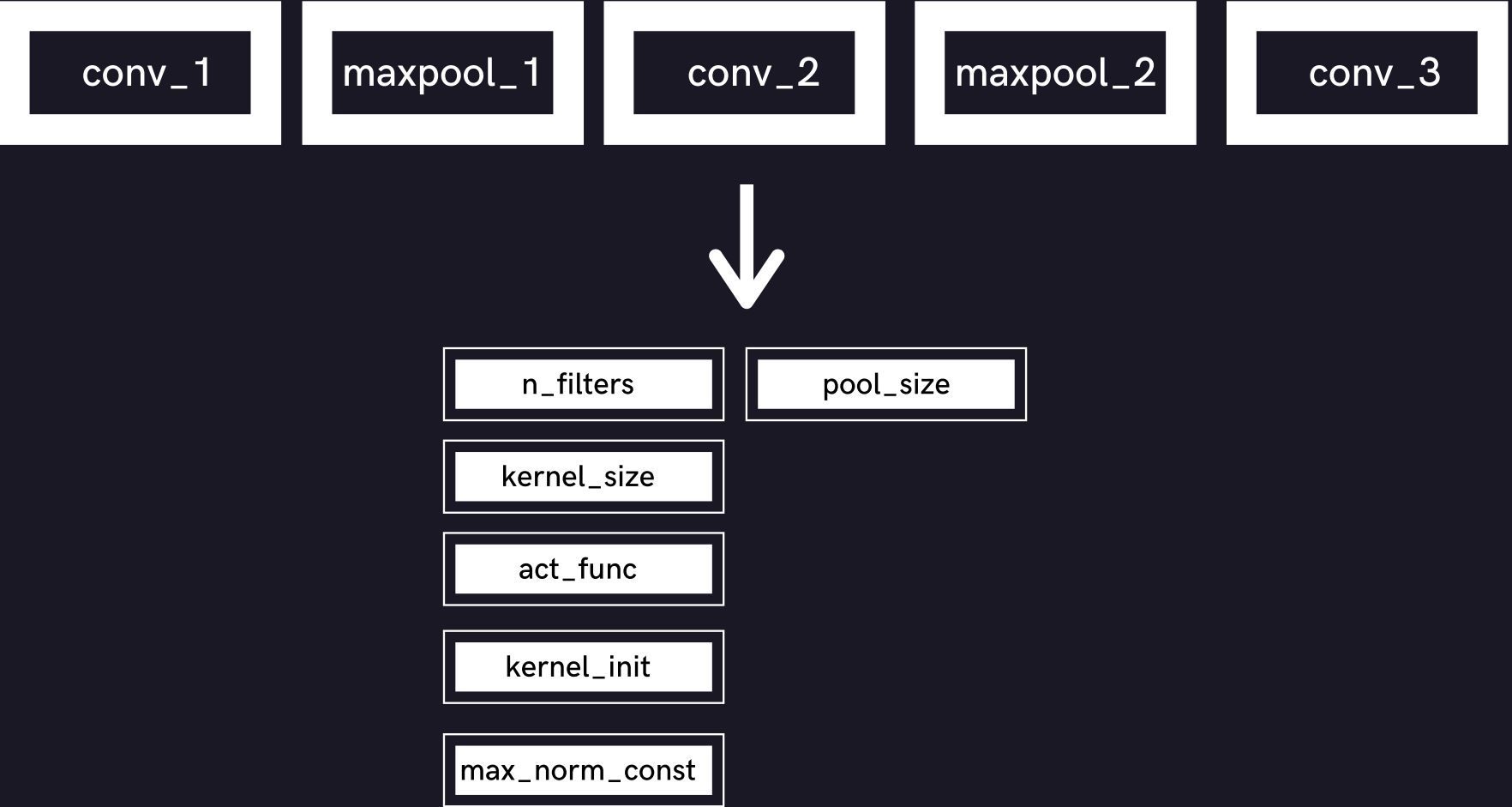
DIFFERENT HYPERPARAMETERS FOR EACH LAYER

SAME HYPERPARAMETERS FOR EACH LAYER

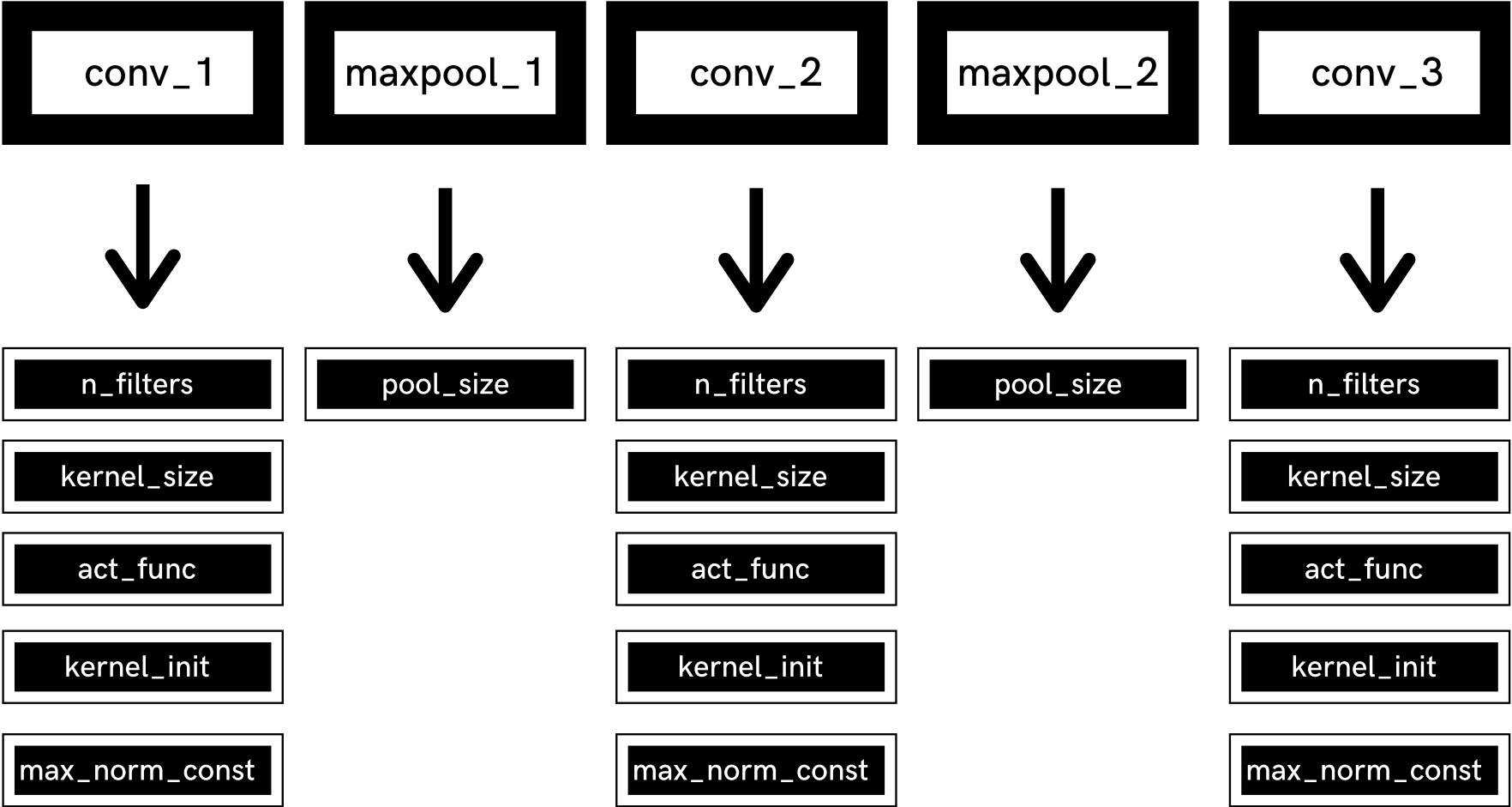


DIFFERENT HYPERPARAMETERS FOR EACH LAYER

SAME HYPERPARAMETERS FOR EACH LAYER



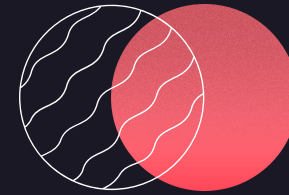
DIFFERENT HYPERPARAMETERS FOR EACH LAYER



RELATED WORK

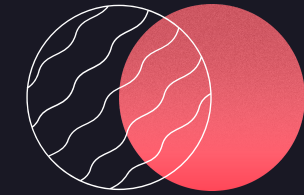


WELL KNOWN APPROACHES



GRID SEARCH

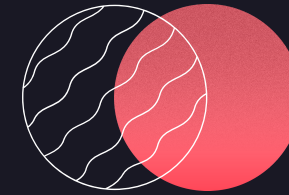
tries all combinations
within the sets of values



RANDOM SEARCH

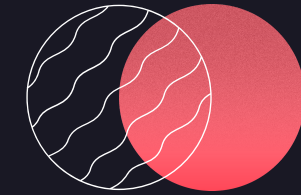
randomly selects a subset of
a combination within the
defined range

WELL KNOWN APPROACHES



GRID SEARCH

tries all combinations
within the sets of values



RANDOM SEARCH

randomly selects a subset of
a combination within the
defined range

NO MECHANISM TO LEARN FROM THE PREVIOUS SEARCH

DEEP NEUROEVOLUTION PAPER

Paper	Year	Optimization Method	Classification Method	Number of Hyperparameters
[1]	2017	EA	CNN	5
[2]	2017	ES	DNN	2
[3]	2018	GA	CNN	6
[4]	2018	GA	CNN	9
[5]	2019	GA	BiLSTM	4
[6]	2019	DE	CNN	2
[7]	2020	GA	CNN	4
Ours	2020	GA	CNN	20

CNN HYPERPARAMETERS AND ARCHITECTURE



CNN HYPERPARAMETERS

GLOBAL HYPERPARAMETERS

those that influence
the overall model

LAYER HYPERPARAMETERS

define the hyperparameters
at the layer level

ARCHITECTURE HYPERPARAMETERS

determine the depth
of the CNN

GLOBAL HYPERPARAMETERS

Hyperparameter	Definition	Range/Values
NE	Number of epochs	Min: 1, Max: 100, Default: 10
BS	Batch size	Min: 32, Max: 256, Default: 32
OP	Optimizer	['sgd', 'rmsprop', 'adagrad', 'adadelat', 'adam', 'adamax', 'nadam'], Default: 'adam'
LR	Learning rate	Min: 1e-4, Max: 1e-2, Default: 1e-4
MO	Momentum	Min: 0.0, Max: 1.0, Default: 0.9

LAYER HYPERPARAMETERS

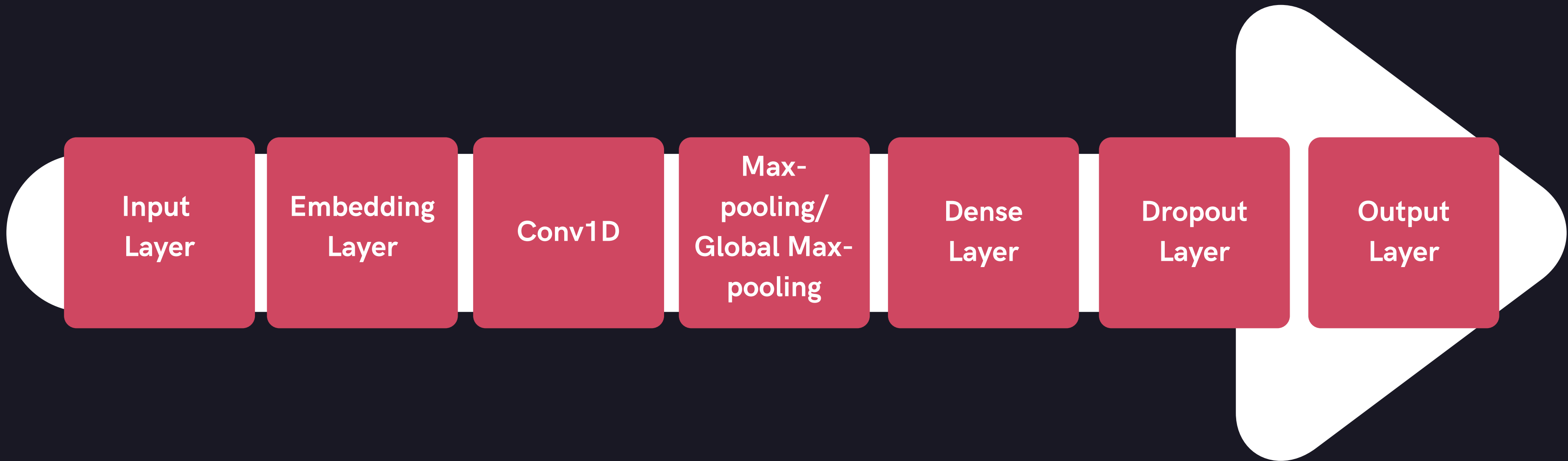
Hyper-parameter	Definition	Range/Values
OD	Output dimension (Embedding)	[50, 100, 200, 300], Default: 100
NF	Number of filters (Convolutional)	Min: 32, Max: 512, Default: 64
KS	Kernel size (Convolutional)	Min: 1, Max: 5, Default: 3
AFC	Activation function (Convolutional)	['relu', 'softmax', 'elu', 'selu', 'softplus', 'softsign', 'tanh', 'sigmoid', 'hard sigmoid', 'linear'], Default: 'relu'
KIC	Kernel initializer (Convolutional)	['zeros', 'ones', 'uniform', 'normal', 'glorot normal', 'glorot uniform', 'he normal', 'he uniform', 'lecunnormal', 'lecun uniform'], Default: 'glorot uniform'
WCC	Max-norm weight constraint (Convolutional)	Min: 1, Max: 5, Default: 3
NN	Number of neurons (Dense)	Min: 1, Max: 30, Default: 1
AFD	Activation function (Dense)	['relu', 'softmax', 'elu', 'selu', 'softplus', 'softsign', 'tanh', 'sigmoid', 'hard sigmoid', 'linear'], Default: 'relu'
KID	Kernel initializer (Dense)	['zeros', 'ones', 'uniform', 'normal', 'glorot normal', 'glorot uniform', 'he normal', 'he uniform', 'lecunnormal', 'lecun uniform'], Default: 'glorot uniform'
WCD	Max-norm weight constraint (Dense)	Min: 1, Max: 5, Default: 3
DR	Dropout rate (Dropout)	Min: 0.0, Max: 1.0, Default: 0.2
PS	Pool size (Max-pooling)	Min: 2, Max: 6, Default: 5
KIO	Kernel initializer (Output)	['zeros', 'ones', 'uniform', 'normal', 'glorot normal', 'glorot uniform', 'he normal', 'he uniform', 'lecunnormal', 'lecun uniform'], Default: 'glorot uniform'

ARCHITECTURE HYPERPARAMETERS

Hyperparameters	Definition	Range/Values
NC	Number of convolutional layers	Min: 1, Max: 15, Default: 1
ND	Number of dense layers	Min: 0, Max: 15, Default: 1

CNN ARCHITECTURE

16



GA-CNN



LOGICAL FLOW

18



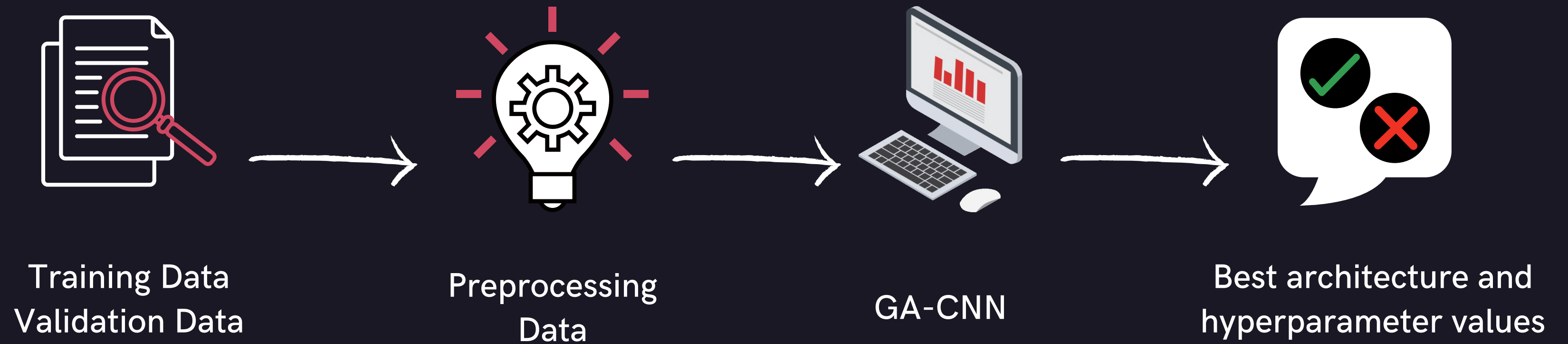
LOGICAL FLOW

18



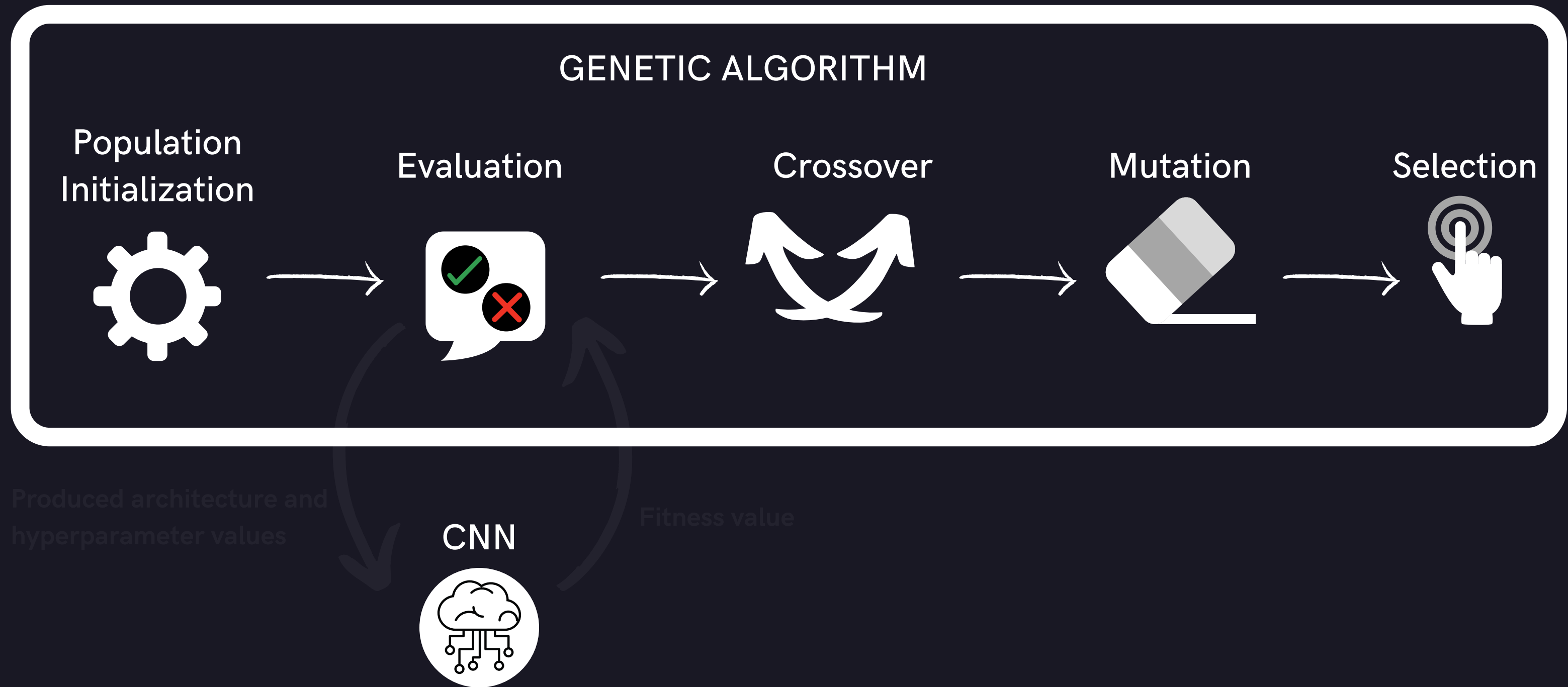
LOGICAL FLOW

18



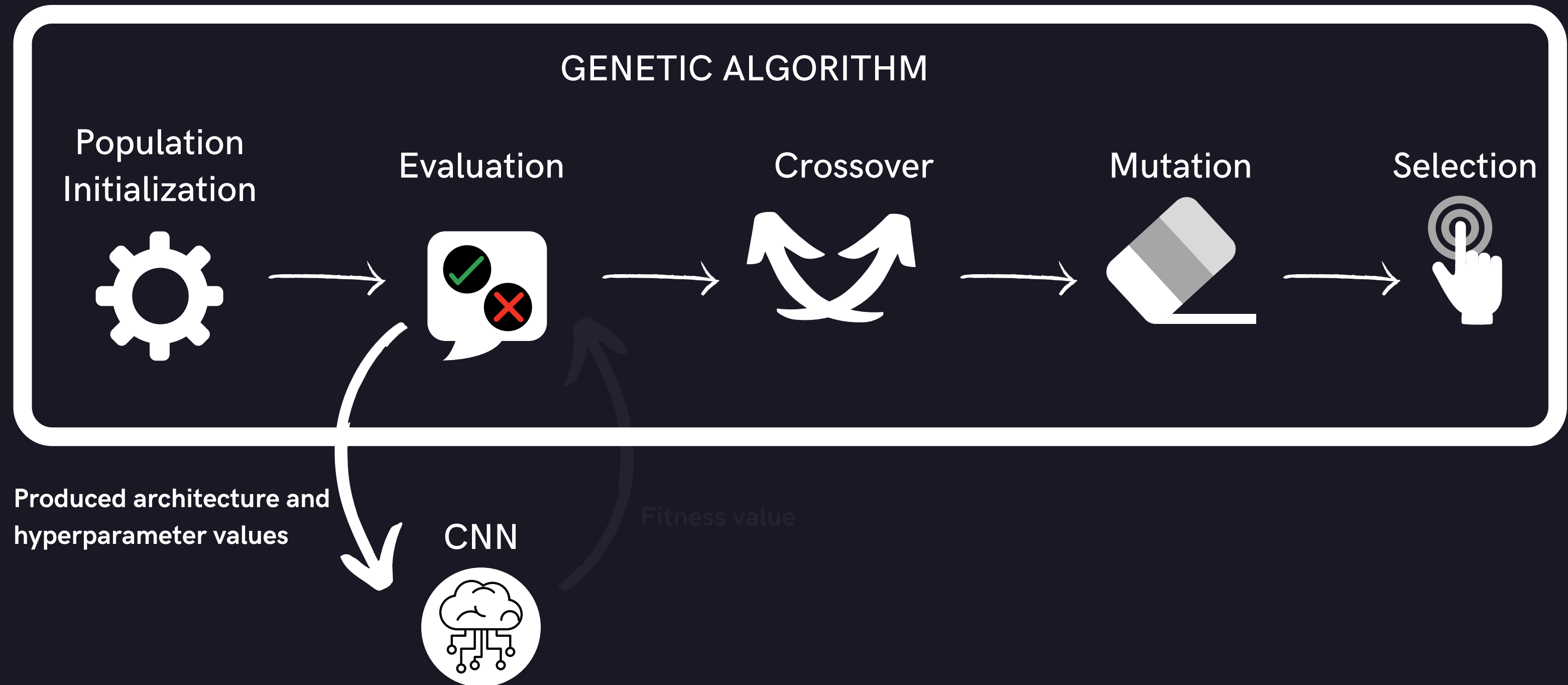
GA-CNN

19



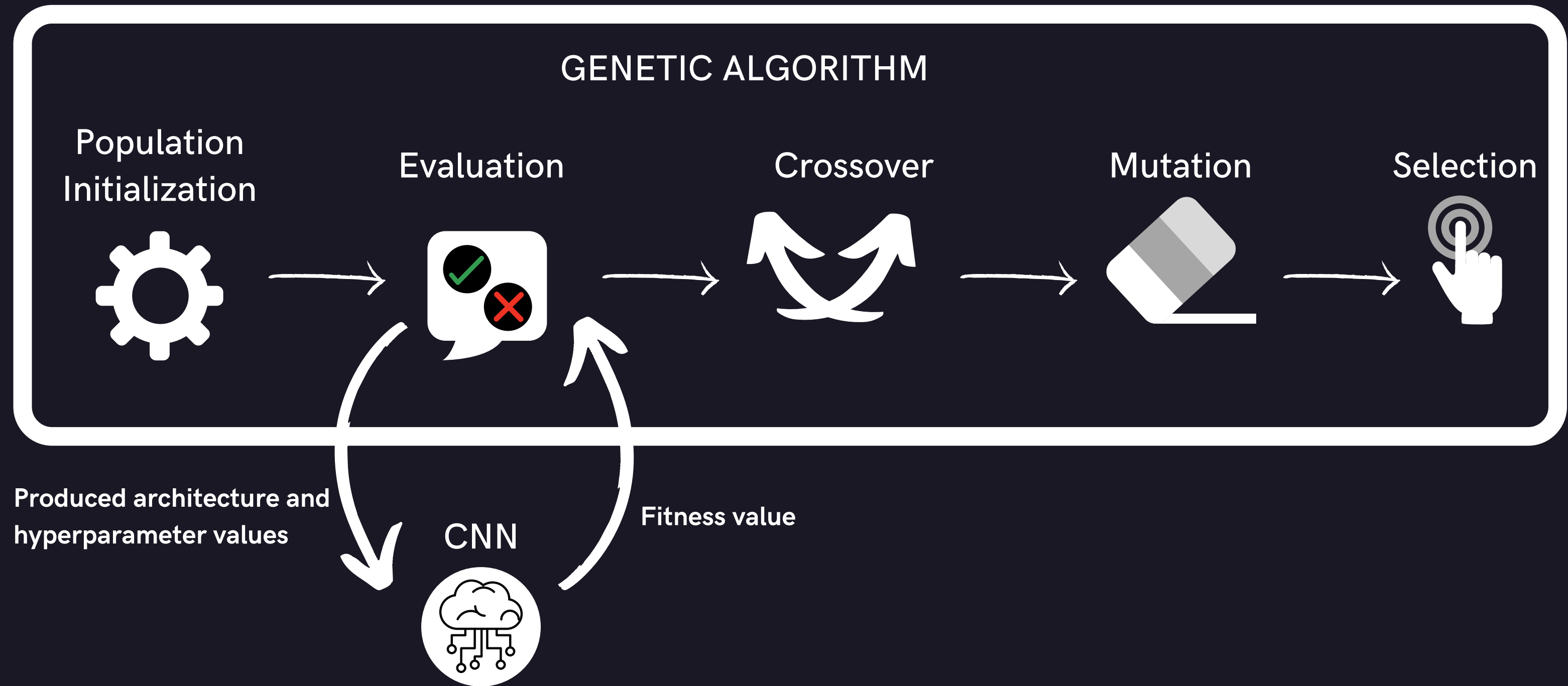
GA-CNN

19



GA-CNN

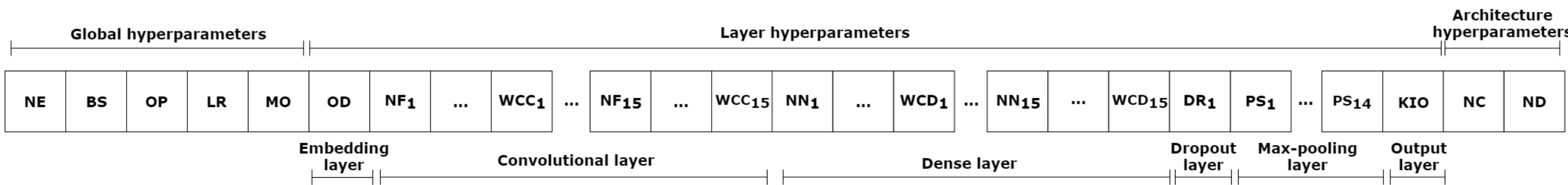
19



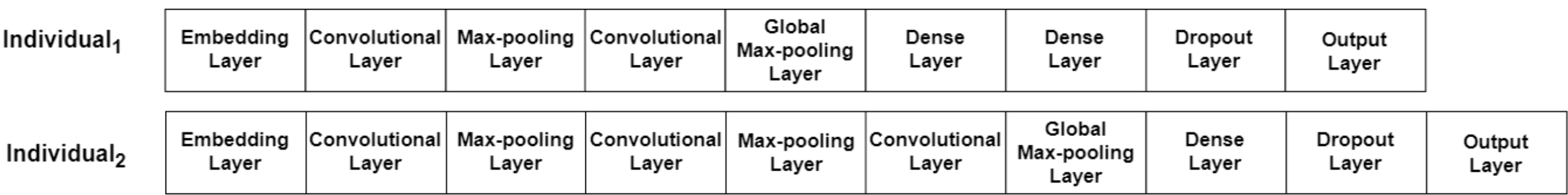
GA-CNN

Population Initialization

Hyperparameter chromosomes



Architecture chromosomes



GA-CNN

21

Evaluation

F1-Score

Crossover

Hyperparameter chromosomes: One-point, two-point, and uniform, which were selected randomly

Architecture chromosomes: One-point crossover with a global max-pooling layer as the cutting point

Mutation

Hyperparameter chromosomes: Uniform Mutation

Architecture chromosomes: Adding or removing a convolutional layer or a dense layer.

Selection

Elitism

RESULTS AND DISCUSSION



DATASET

23

DISASTER TWEET DATASET

Supervised text classification

Two classes: relevant and not relevant

Training : Validation : Testing = 49 : 21 : 30

GA PARAMETER

24

Parameter	Description	Value
CR	Crossover rate	0.8
MR	Mutation rate	0.2
Ngen	Number of generations	30
Npop	Number of populations	30

COMPARISON

25

TRAINING

Find the best architecture and
hyperparameters from GA-CNN

TESTING

Test the best architecture and
hyperparameters from GA-CNN
using test dataset

TRAINING

Hyperparameter	Experiment						
	1	2	3	4	5	6	7
NE	✓	✓	✓	✓	✓	✓	✓
BS	✓	✓	✓	✓	✓	✓	✓
OP	✓	✓	✓	✓	✓	✓	✓
LR	✓	✓	✓	✓	✓	✓	✓
MO						✓	✓
OD		✓*	✓**	✓*	✓**	✓*	✓**
NF		✓*	✓**	✓*	✓**	✓*	✓**
KS		✓*	✓**	✓*	✓**	✓*	✓**
AFC				✓*	✓**	✓*	✓**
KIC		✓*	✓**	✓*	✓**	✓*	✓**
WCC				✓*	✓**	✓*	✓**
NN				✓*	✓**	✓*	✓**
AFD				✓*	✓**	✓*	✓**
KID				✓*	✓**	✓*	✓**
WCD				✓*	✓**	✓*	✓**
DR						✓	✓
PS						✓*	✓**
KIO						✓	✓
NC	✓	✓	✓	✓	✓	✓	✓
ND	✓	✓	✓	✓	✓	✓	✓
Number of hyperparameters	6	10	10	15	15	20	20
Number of optimized hyperparameters	6	10	66	15	141	20	159

Notes: *) Same values for all layers. **) Different values for all layers.

TRAINING

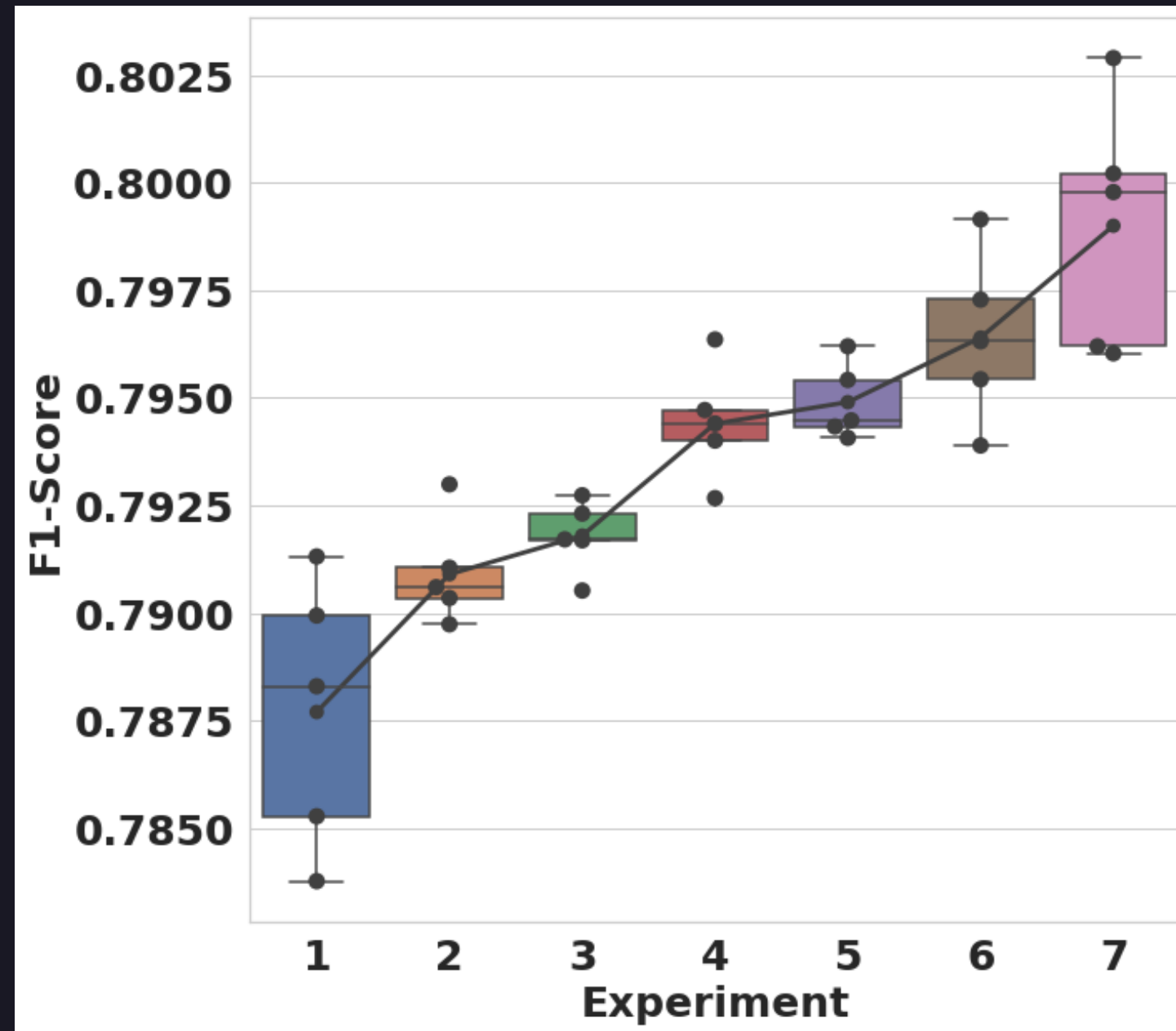
27

Method	Best F1-Score							Average F1-Score						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
Grid Search	0.7357	-	-	-	-	-	-	0.0016	-	-	-	-	-	-
Random Search	0.7701	0.7849	0.7831	0.7836	0.7848	0.7875	0.7871	0.1492	0.4725	0.4535	0.4387	0.4172	0.3680	0.3726
TPE [9]	0.7867	0.7905	0.7924	0.7913	0.7905	0.7934	0.7948	0.7833	0.7873	0.7880	0.7893	0.7876	0.7889	0.7903
GA-CNN	0.7913	0.7930	0.7927	0.7964	0.7962	0.7991	0.8029	0.7877	0.7909	0.7918	0.7944	0.7949	0.7964	0.7990
TPOT [10]	0.6535							0.6455						

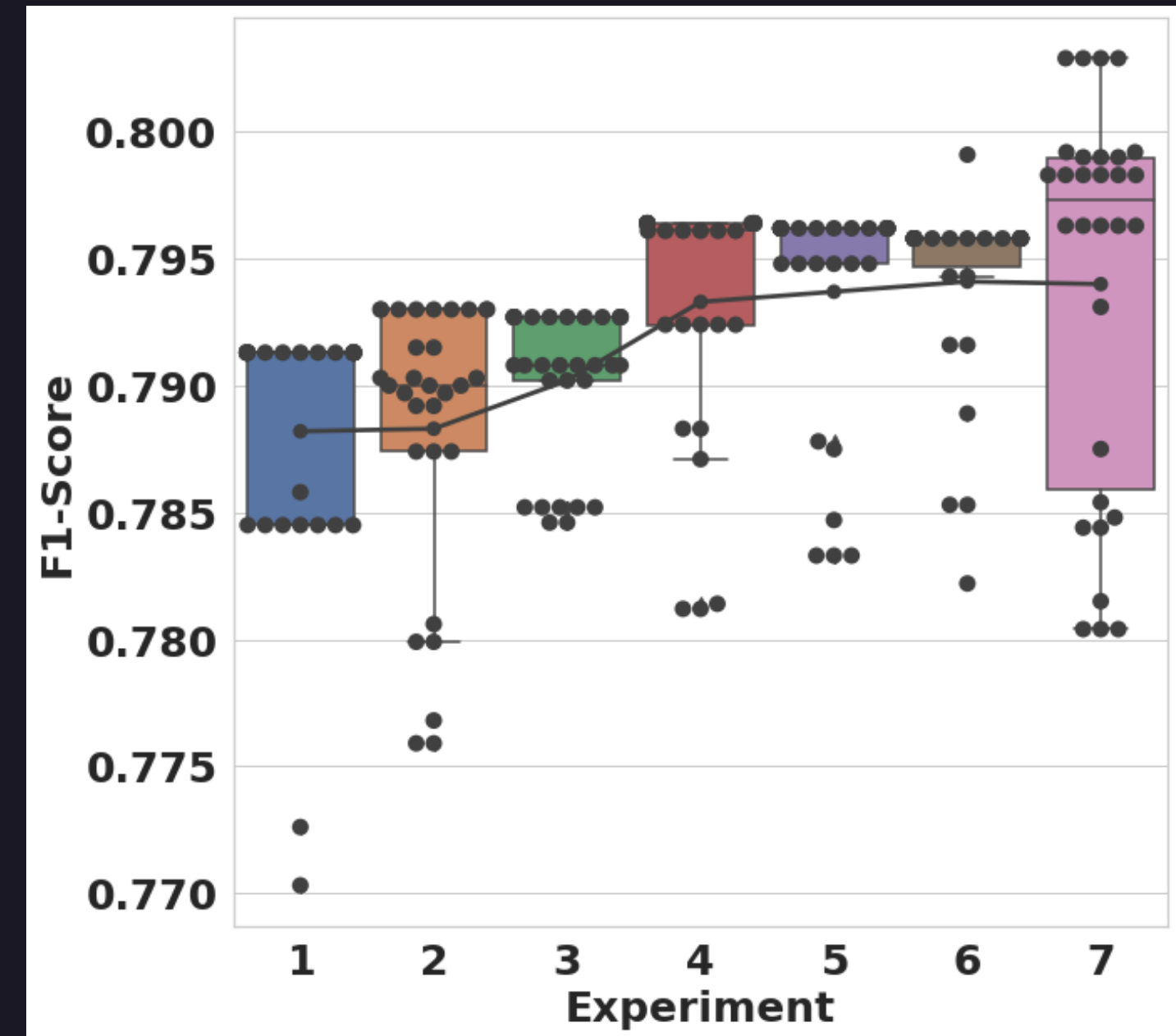
Method	Average Time (s)						
	1	2	3	4	5	6	7
Grid Search	24268	-	-	-	-	-	-
Random Search	14473	18746	17910	17399	17203	21462	21581
TPE [9]	22009	26843	25440	26443	25661	31302	31146
GA-CNN	18301	21573	20347	17983	17909	26457	21833
TPOT [10]	21186						

TRAINING

28



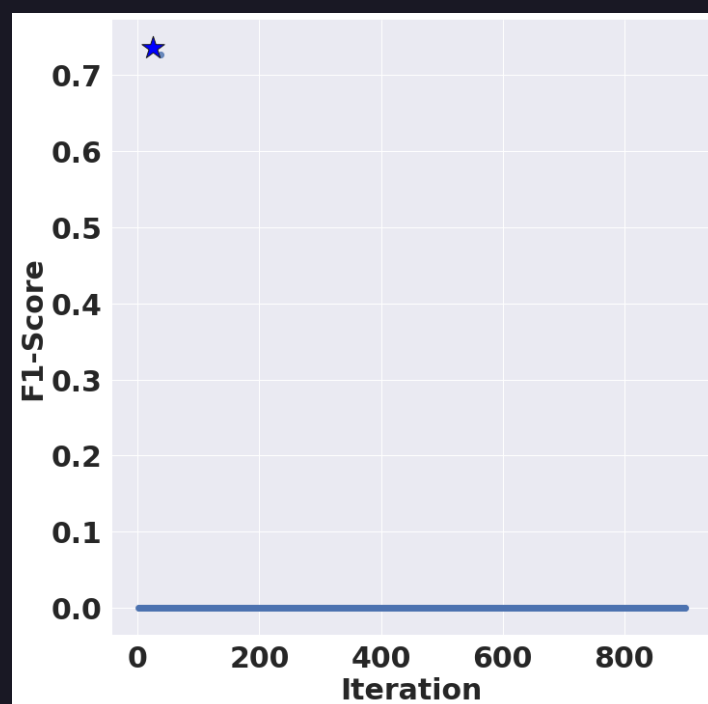
All the final F1-Scores of GA-CNN from 5 runs



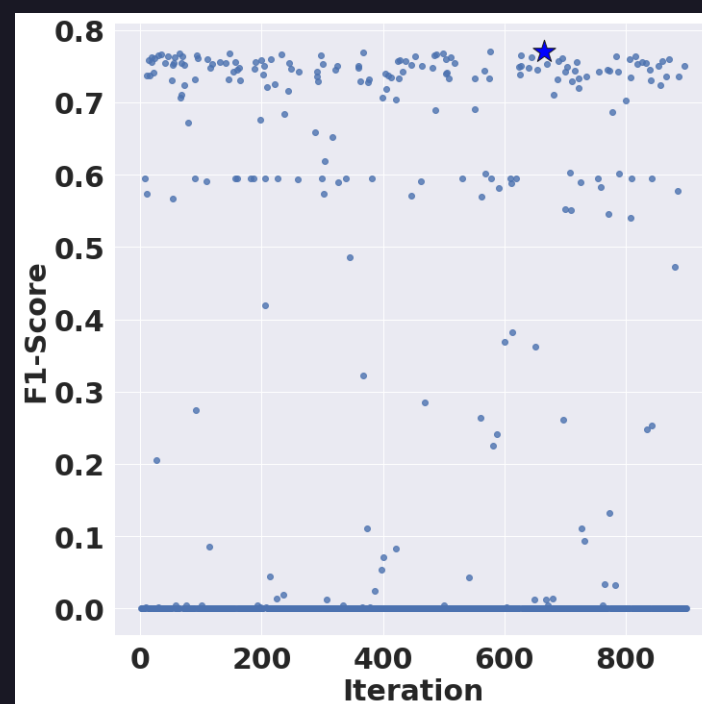
The best GA-CNN results out of 5 runs for 30 generations

TRAINING

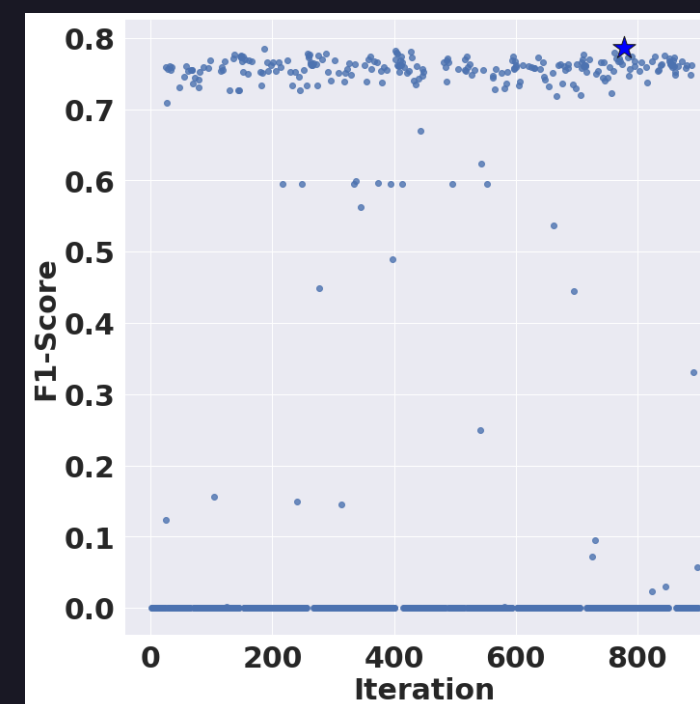
29



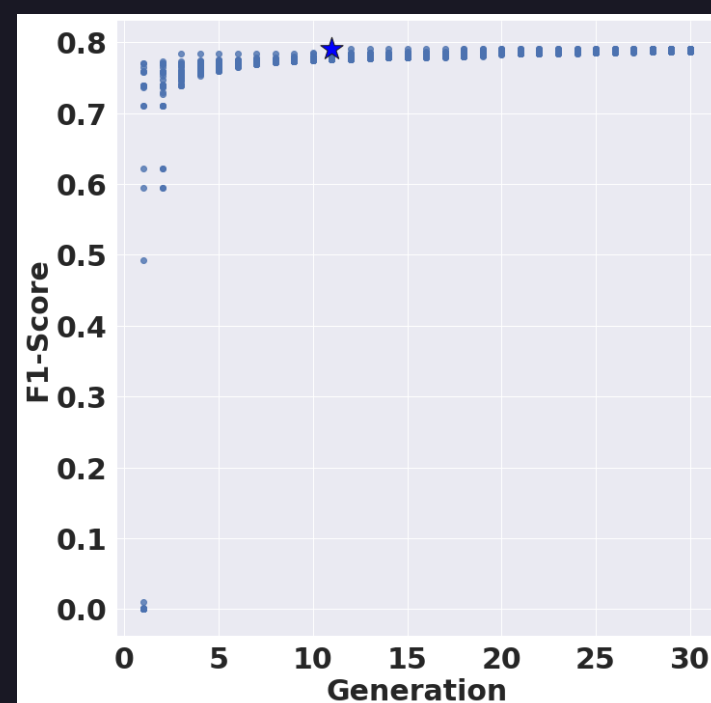
Grid Search



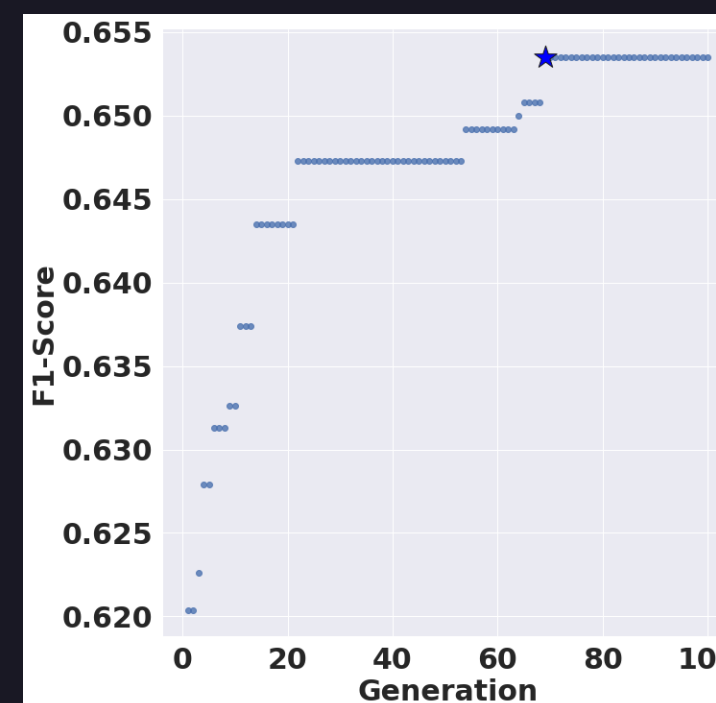
Random Search



TPE



GA-CNN



TPOT

TRAINING

30

Global Hyper-parameters NE: 30 BS: 187 OP: adamax LR: 0.0076627 MO: 0.9*	Input Layer	Embedding Layer OD: 100*	Convolutional Layer NF: 64* KS: 3* AFC: relu* KIC: glorot uniform* WCC: 3*	Global Max-pooling Layer	Dropout Layer DR: 0.2*	Output Layer KIO: glorot uniform*			
Global Hyper-parameters NE: 91 BS: 73 OP: adamax LR: 0.0088001 MO: 0.9*	Input Layer	Embedding Layer OD: 100*	Convolutional Layer NF: 429 KS: 1 AFC: softplus KIC: glorot uniform* WCC: 3*	Global Max-pooling Layer	Dense Layer NN: 25 AFD: relu* KID: glorot uniform* WCD: 3*	Dropout Layer DR: 0.2*	Output Layer KIO: glorot uniform*		
Global Hyper-parameters NE: 38 BS: 146 OP: adamax LR: 0.0012272 MO: 0.9*	Input Layer	Embedding Layer OD: 100*	Convolutional Layer NF: 219 KS: 1 AFC: sigmoid KIC: glorot uniform* WCC: 3*	Max-pooling Layer PS: 5*	Convolutional Layer NF: 445 KS: 2 AFC: relu KIC: glorot uniform* WCC: 3*	Global Max-pooling Layer	Dense Layer NN: 29 AFD: relu* KID: glorot uniform* WCD: 3*	Dropout Layer DR: 0.2*	Output Layer KIO: glorot uniform*
Global Hyper-parameters NE: 29 BS: 104 OP: adamax LR: 0.00723 MO: 0.9*	Input Layer	Embedding Layer OD: 100*	Convolutional Layer NF: 74 KS: 1 AFC: linear KIC: ones WCC: 5	Global Max-pooling Layer	Dense Layer NN: 17 AFD: hard sigmoid KID: normal WCD: 4	Dropout Layer DR: 0.2*	Output Layer KIO: glorot uniform*		
Global Hyper-parameters NE: 90 BS: 69 OP: adamax LR: 0.0060712 MO: 0.9*	Input Layer	Embedding Layer OD: 100*	Convolutional Layer NF: 70 KS: 1 AFC: hard sigmoid KIC: lecun normal WCC: 5	Global Max-pooling Layer	Dense Layer NN: 29 AFD: hard sigmoid KID: normal WCD: 2	Dropout Layer DR: 0.2*	Output Layer KIO: glorot uniform*		
Global Hyper-parameters NE: 83 BS: 130 OP: adamax LR: 0.0021904 MO: 0.7933697	Input Layer	Embedding Layer OD: 200	Convolutional Layer NF: 144 KS: 2 AFC: selu KIC: glorot normal WCC: 1	Global Max-pooling Layer	Dense Layer NN: 14 AFD: sigmoid KID: lecun uniform WCD: 5	Dropout Layer DR: 0.7381949	Output Layer KIO: zeros		
Global Hyper-parameters NE: 66 BS: 153 OP: adamax LR: 0.0024118 MO: 0.4915175	Input Layer	Embedding Layer OD: 200	Convolutional Layer NF: 256 KS: 1 AFC: elu KIC: ones WCC: 4	Global Max-pooling Layer	Dense Layer NN: 25 AFD: softsign KID: he uniform WCD: 2	Dropout Layer DR: 0.1005055	Output Layer KIO: uniform		

TESTING

Method	Best F1-Score							Average F1-Score						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
Grid Search	0.0000	-	-	-	-	-	-	0.0000	-	-	-	-	-	-
Random Search	0.0000	0.7521	0.7536	0.7512	0.7495	0.7644	0.7654	0.0000	0.7313	0.7461	0.7410	0.7374	0.7267	0.6566
TPE [9]	0.0000	0.7489	0.7519	0.7544	0.7462	0.7669	0.7680	0.0000	0.6442	0.7449	0.7477	0.6749	0.7605	0.7597
GA-CNN	0.0000	0.7545	0.7548	0.7593	0.7563	0.7728	0.7747	0.0000	0.7388	0.7467	0.7435	0.7421	0.7585	0.7625
TPOT [10]	0.6372							0.6286						

Method	Best F1-Score
Grid Search	0.0000
Random Search	0.7654
TPE [9]	0.7680
GA-CNN	0.7747
TPOT [10]	0.6372
CNN [11]	0.0000
CNN [12]	0.7654
LSTM [13]	0.7680
CNN and LSTM [14]	0.7747
BERT [15]	0.6372

WILCOXON SIGNED-RANK TEST FOR GA-CNN

32

Experiment	Training							Testing						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1	x	0.18	0.043	0.018	0.018	0.018	0.018	x	0.028	0.028	0.028	0.028	0.028	0.028
2		x	0.068	0.018	0.018	0.018	0.018		x	0.028	0.028	0.027	0.028	0.028
3			x	0.043	0.043	0.043	0.043			x	0.042	0.028	0.028	0.028
4				x	0.068	0.043	0.043				x	0.028	0.028	0.028
5					x	0.18	0.043					x	0.028	0.028
6						x	0.043						x	0.028
7							x							x

CONCLUSIONS AND FUTURE WORK



CONCLUSIONS

34

The results from seven experiments showed that a larger number of hyperparameters and layer-specific hyperparameter values are important.

The Wilcoxon signed-rank test also exposed the superiority of the GA-CNN with a larger number of hyperparameters.

The GA-CNN produced higher F1-Scores than the other methods in training and testing

The optimization of the output dimension and max-norm weight constraint hyperparameters has a notable impact on the search process, especially when different hyperparameter values for each layer are investigated.

FUTURE WORK

35

Dynamic CNN architecture design to achieve higher adaptability over different problems.

It is interesting to select important hyperparameters automatically and optimize them.

The CR and MR values should be able to adapt based on the present situation.

Further comparisons should be investigated to gain more insight from the proposed approach.

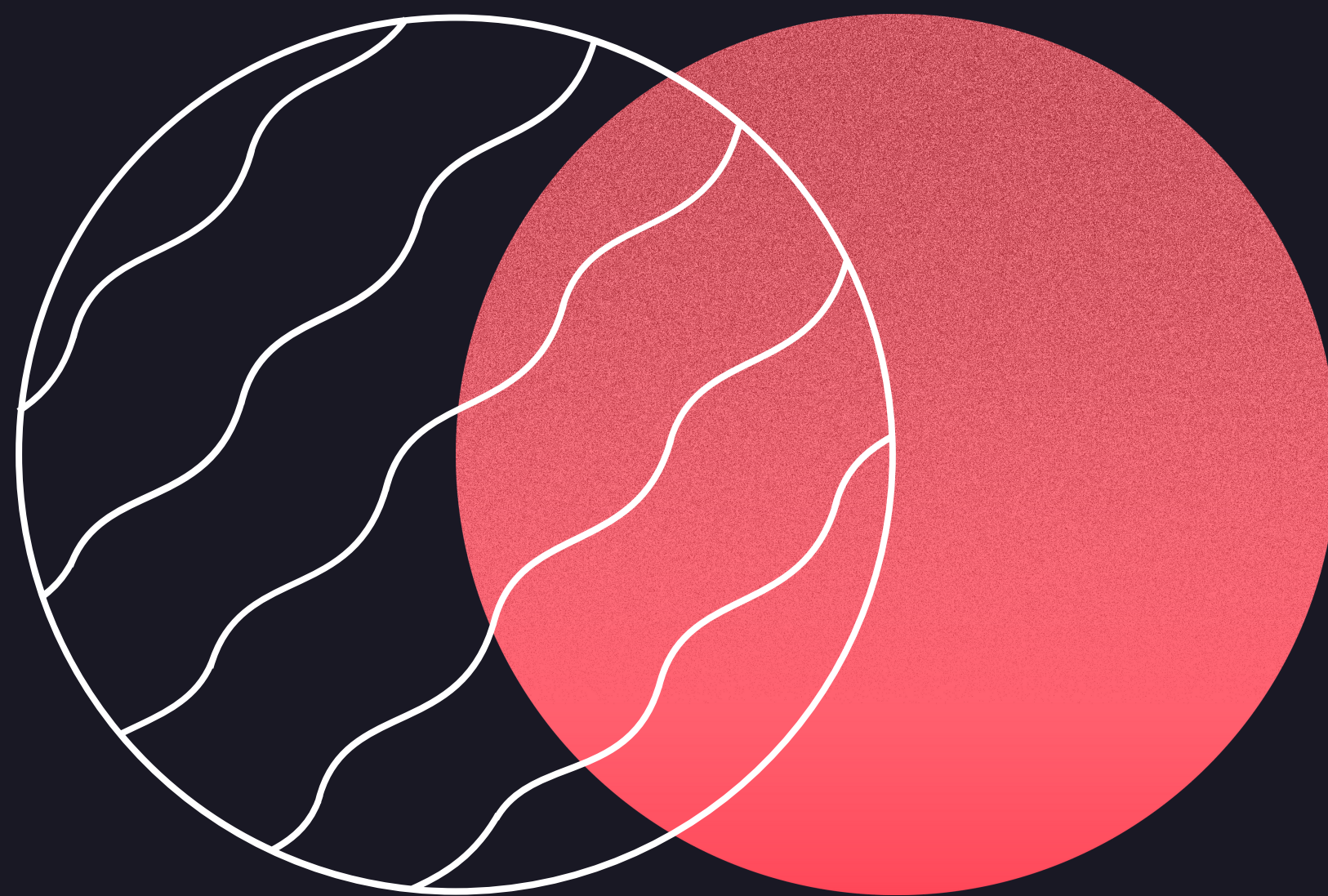
REFERENCES

- [1] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *Proceedings of the 34th International Conference on Machine Learning- Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 2902–2911.
- [2] P. Vidnerova and R. Neruda, "Evolution Strategies for Deep Neural Network Models Design," in *Information Technologies – Applications and Theory Conference, ITAT 2017*, 2017, pp. 159–166. [Online]. Available: <http://ceur-ws.org/Vol-1885/159.pdf>
- [3] G. C. Felbinger, "Optimal CNN Hyperparameters for Object Detection on NAO Robots," Master's thesis, Technische Universität Hamburg-Harburg, 2018. [Online]. Available: https://hulks.de/files/MA_Georg-Felbinger.pdf
- [4] T. Hinz, N. Navarro-Guerrero, S. Magg, and S. Wermter, "Speeding up the hyperparameter optimization of deep convolutional neural networks," *International Journal of Computational Intelligence and Applications*, vol. 17, no. 02, p. 1850008, 2018.
- [5] N. Bansal, A. Sharma, and R. Singh, "An Evolving Hybrid Deep Learning Framework for Legal Document Classification," *Ingenierie des systèmes d'information*, vol. 24, no. 4, pp. 425–431, oct 2019. [Online]. Available: <http://www.iieta.org/journals/isi/paper/10.18280/isi.240410>
- [6] A. Dahou, M. A. Elaziz, J. Zhou, and S. Xiong, "Arabic Sentiment Classification Using Convolutional Neural Network and Differential Evolution Algorithm," *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–16, feb 2019. [Online]. Available: <https://www.hindawi.com/journals/cin/2019/2537689/>
- [7] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification," *IEEE Transactions on Cybernetics*, pp. 1–15, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9075201/>

REFERENCES

37

- [8] Kaggle, "Real or Not? NLP with Disaster Tweets." [Online]. Available: <https://www.kaggle.com/c/nlp-getting-started>
- [9] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ser. ICML'13. JMLR.org, 2013, p. I-115-I-123.
- [10] R. S. Olson and J. H. Moore, "TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning," in Automated Machine Learning: Methods, Systems, Challenges. Cham: Springer International Publishing, 2019, pp. 151-160. [Online]. Available: http://link.springer.com/10.1007/978-3-030-05318-5_8
- [11] E. Green, "Pytorch CNN Text," 2020. [Online]. Available: <https://www.kaggle.com/eliasgreen/pytorch-cnn-text>
- [12] L. Hofman, "Disaster or not? Glove embedded CNN," 2020. [Online]. Available: <https://www.kaggle.com/luukhofman/disaster-or-not-glove-embedded-cnn>
- [13] Vivekpandian, "Basic LSTM," 2020. [Online]. Available: <https://www.kaggle.com/vivekpandain/basic-lstm>
- [14] D. Karyakin, "CNN and LSTM," 2020. [Online]. Available: <https://www.kaggle.com/dmikar/cnn-and-lstm>
- [15] P. Kumar, "BERT fastai," 2020. [Online]. Available: <https://www.kaggle.com/kpriyanshu256/bert-fastai>



THANK YOU