Classification of Actionable Gene Mutation Using N.I.A

Group Members

K16-3974	Sohail Zia
K16-3966	Sanif Ali Momin
K16-3986	Ahmed Raza

Contents

Problem Statement

“ Classification of Actionable Genetic Mutations using Nature Inspired Algorithmic Approach ”

Deep learning is faced with many limitations, but not limited to:

- Backprop algorithm scales exponentially with increased complexity of the problem.

- Not robust to changes of network parameters, such as #hidden layers, activation functions, learning rate, etc.

- Local minima trap (*though a rare problem[1]*).

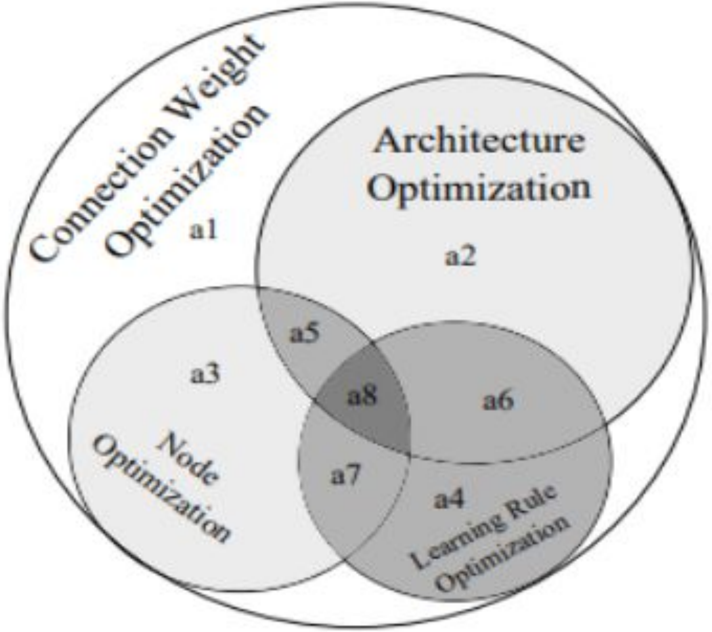
- Gradient flow issues with Deep neural networks.

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, p. 436, 2015.

Incorporating NIA in Neural Networks

- Advantage of two major components of metaheuristics
- Optimum Weights can be realized
- Can find near optimal solutions at a minimal computational cost
- No need for differentiable activation functions or loss functions.

Spectrum of Neural Network Components Optimization

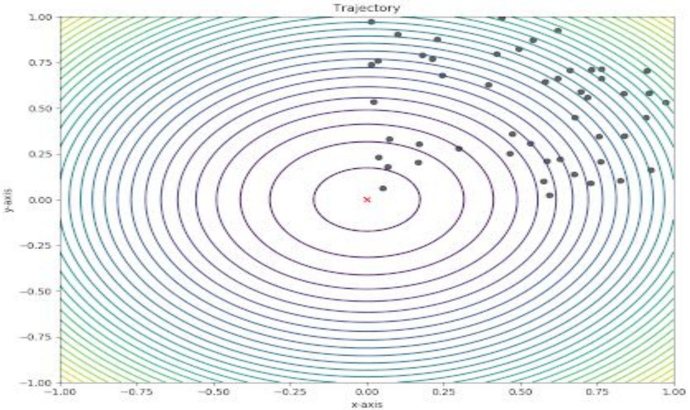


Particle Swarm Optimization

- A swarm of particles communicate with one another using search directions.

- During each iteration, each particle updates its position according to its previous history (local best position) and also its neighbors history (global best position).

- Every particle is composed of 3 vectors:
  (1) x-vector (2) pbest-vector (3) v-vector



$$x_i = x_i + v_{i+1}$$

$$v_{i+1} = w_i + C_1.R_1.(p_i^{best} - x) + C_2.R_2.(p_i^{global} - x)$$

Our Proposed Work

- Replaced Gradient Descent with Particle Swarm Optimization

- We tried different variants of PSO, but so far only one worked. This variant uses an additional formula that linearly decreases the inertia weight during training. [1]

- Comparison of PSO trained models vs. traditional models.

[1] J. Xin, G. Chen, and Y. Hui, A particle swarm optimizer with multistage linearly-decreasing inertia weight in Proc. Int. Joint Conf. Comput. Sci. Optim. (CSO).

Dataset

Kaggle competition: Personalized Medicine: Redefining Cancer Treatment.

Two Stage Competition, provided:

1- Training variant

2- Training Text

3- Test variant

4- Test Text

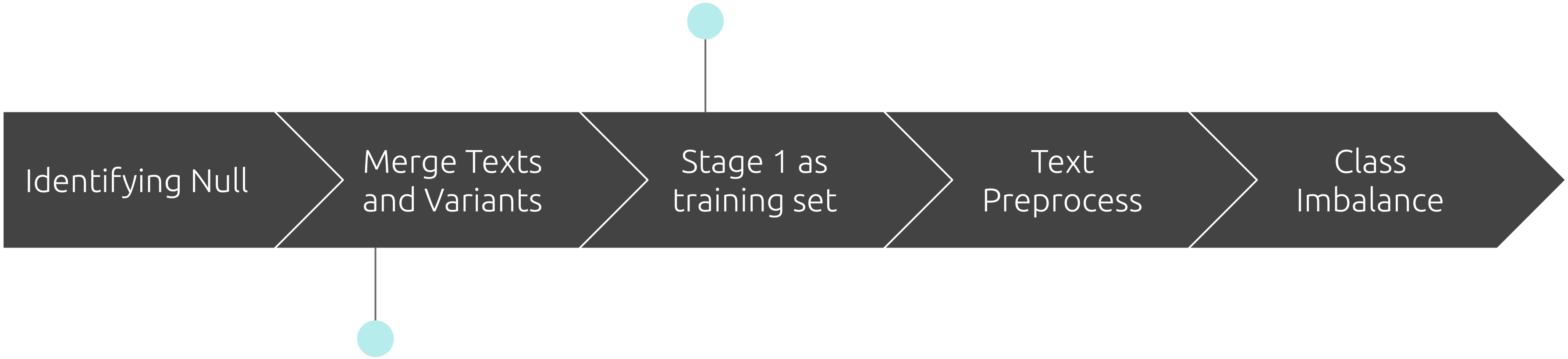
5- Stage 1 filtered solution

6- Stage2 test text

7- Stage2 test variant

Preprocessing

Use Stage 1 filtered
solution as training

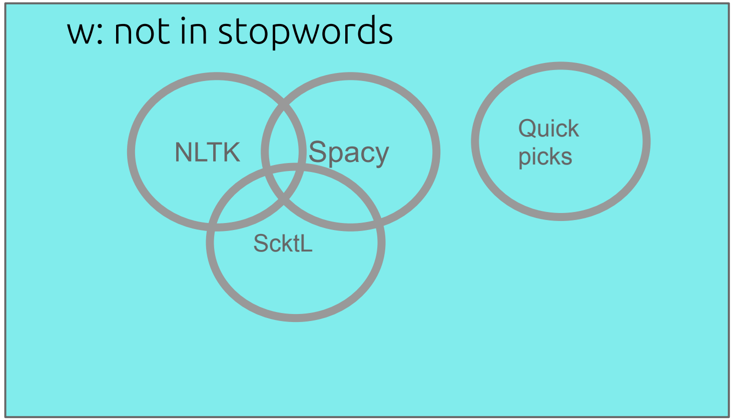


Create a single
dataframe for each
stage

Text Preprocess

Maximum document length (before preprocessing): 76708

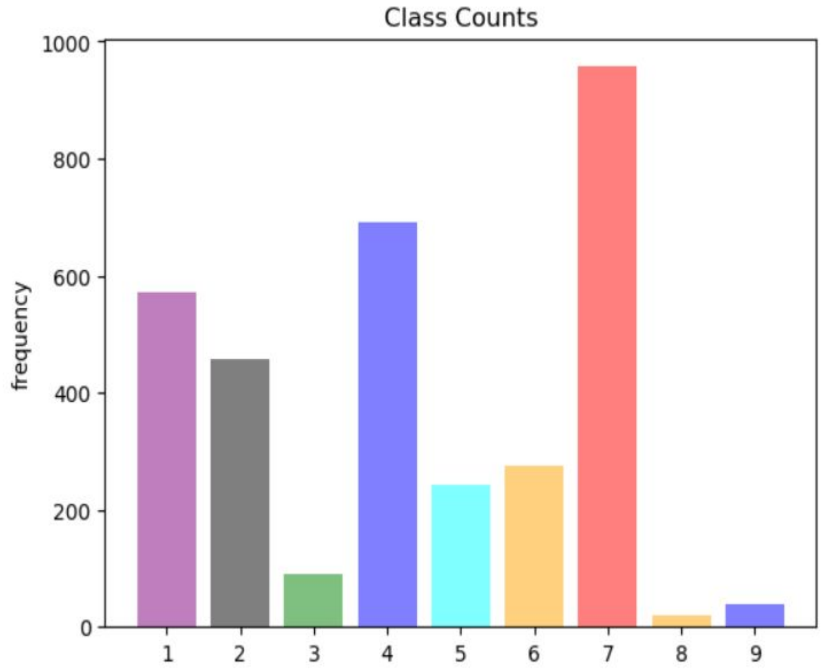
Maximum document length (after preprocessing): 47557



Class Counts

```
count_dict.items()
```

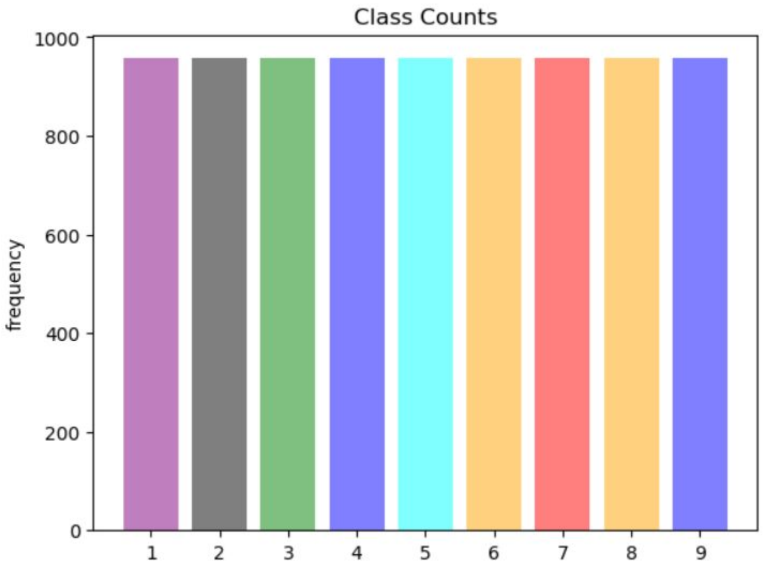
```
dict_items([(1, 573), (2, 458), (3, 90), (4, 690), (5, 242), (6, 275), (7, 957), (8, 19), (9, 38)])
```



SMOTE

```
count_dict.items()
```

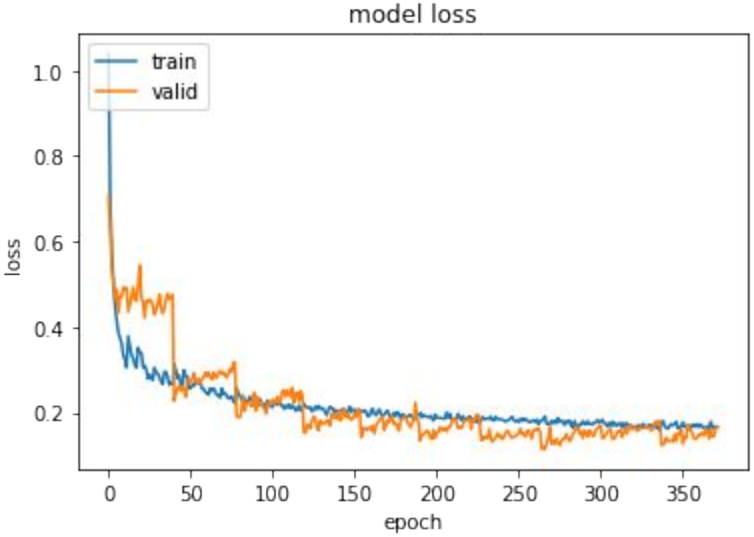
```
dict_items([(1, 957), (2, 957), (3, 957), (4, 957), (5, 957), (6, 957), (7, 957), (8, 957), (9, 957)])
```



Baseline Model Summary

Layers	Units	Parameters
GRU_1	250	375750
GRU_2	250	375750
GRU_3	250	375750
Flatten	200	0
Dense	9	1809
Total Parameters		**1,023,909**

Conventional GRU



Variants

1- First PSO equation

$$x_{i,d}(it+1) = x_{i,d}(it) + v_{i,d}(it+1) \tag{1}$$

$$\begin{aligned} v_{i,d}(it+1) &= v_{i,d}(it) \\ &+ C_1 * Rnd(0,1) * [pb_{i,d}(it) - x_{i,d}(it)] \\ &+ C_2 * Rnd(0,1) * [gb_d(it) - x_{i,d}(it)] \end{aligned} \tag{2}$$

Variants

2- PSO with Constant Inertia

velocity of particle i at time k+1

$$\mathbf{v}_{k+1}^{i} = \underbrace{w\,\mathbf{v}_k^i}_{\text{current motion}} + c_1\, rand\, \underbrace{\frac{\left(\mathbf{p}^i - \mathbf{x}_k^i\right)}{\Delta t}}_{\text{particle memory influence}} + c_2\, rand\, \underbrace{\frac{\left(\mathbf{p}_k^g - \mathbf{x}_k^i\right)}{\Delta t}}_{\text{swarm influence}}$$

inertia factor
range: 0.4 to 1.4

self confidence
range: 1.5 to 2

swarm confidence
range: 2 to 2.5

Variants

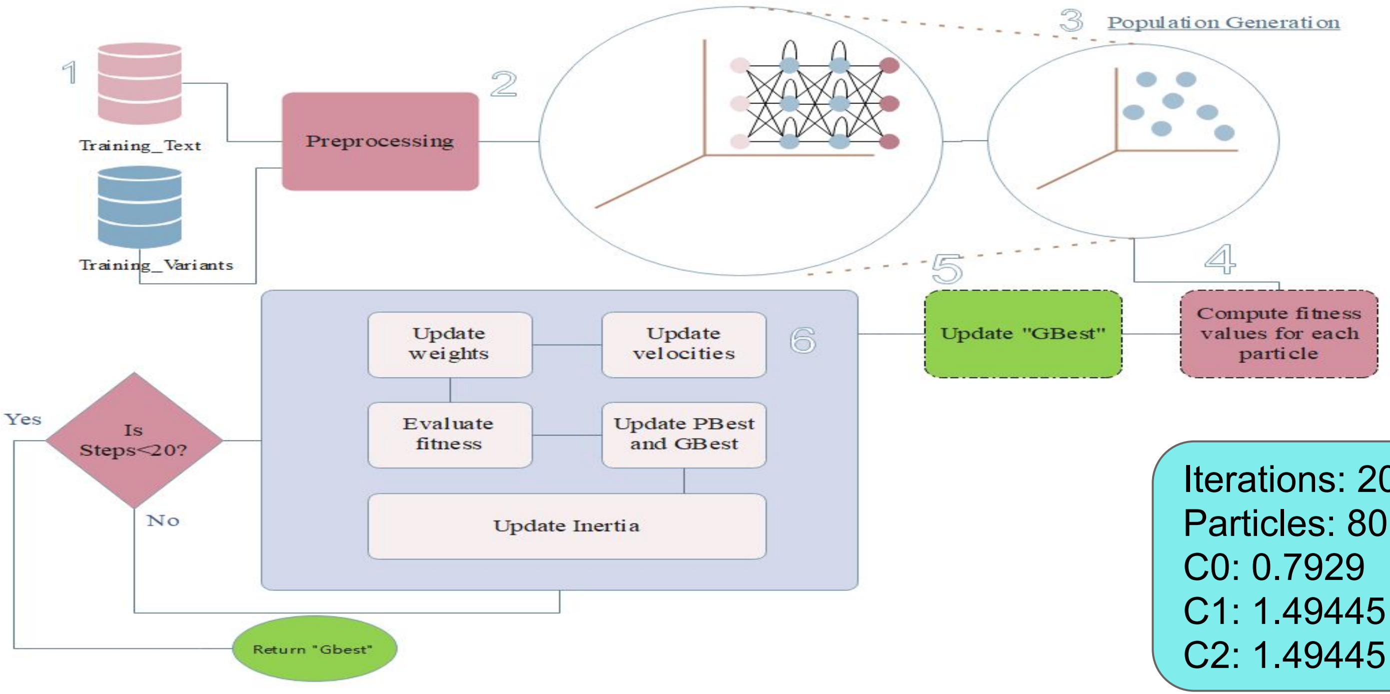
3- Linear Decreasing Inertia Weight

$$w_k = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times k$$

Wmax = 0.9
Wmin = 0.4

Experimental Setup

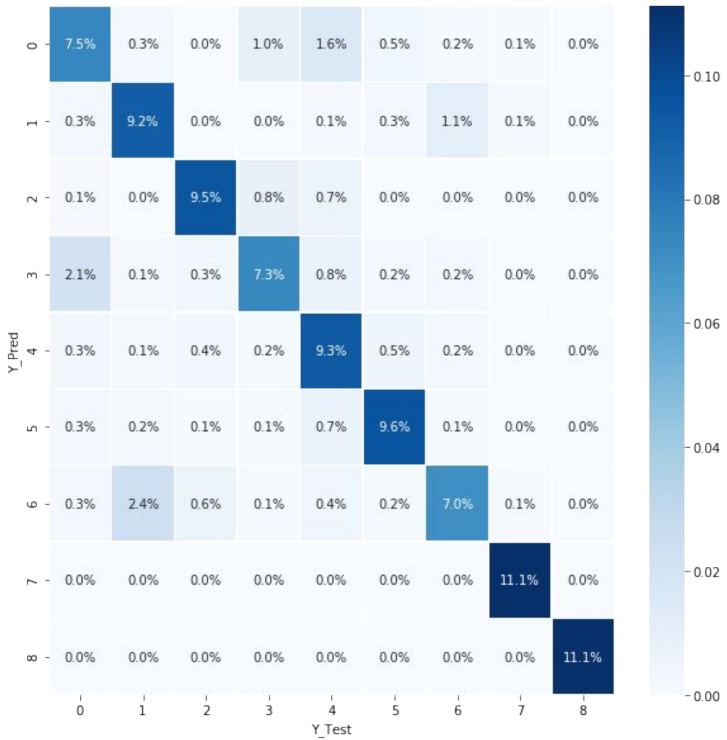
Holistic View of PSO_GRU



Iterations: 20
Particles: 80
C0: 0.7929
C1: 1.49445
C2: 1.49445

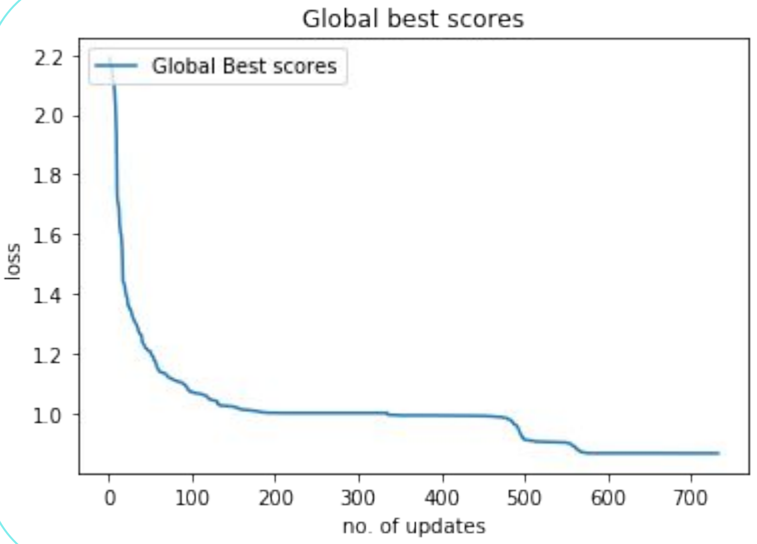
Results

Confusion Matrix and Classification report

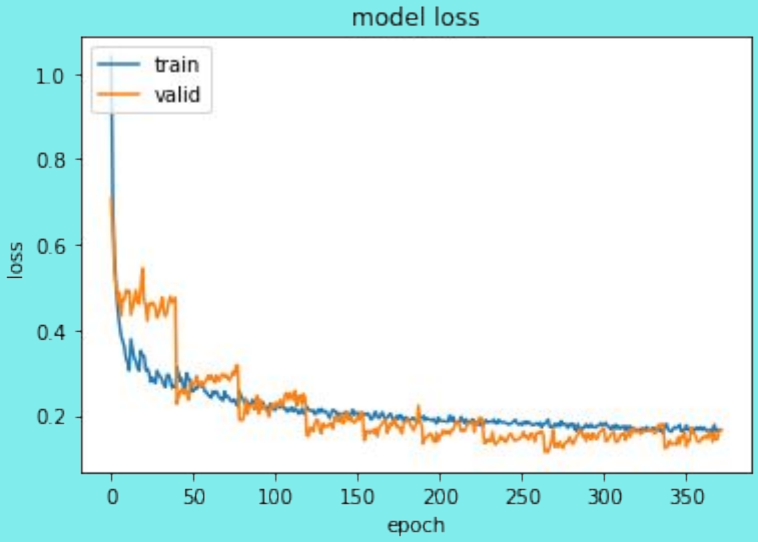


	precision	recall	f1-score	support
class 0	0.69	0.67	0.68	319
class 1	0.74	0.82	0.78	319
class 2	0.86	0.85	0.86	319
class 3	0.77	0.66	0.71	319
class 4	0.68	0.84	0.75	319
class 5	0.85	0.87	0.86	319
class 6	0.79	0.63	0.70	319
class 7	0.97	1.00	0.99	319
class 8	0.99	1.00	1.00	319
accuracy			0.82	2871
macro avg	0.82	0.82	0.81	2871
weighted avg	0.82	0.82	0.81	2871

PSO_GRU Global best scores



Conventional GRU



...Kaggle Score

	Public Score (tested on 40%)	Private Score (tested on 60%)
GRU	1.48	4.37
PSO-GRU	2.48	**2.32**

Discussion

Challenges of PSO to represent in NN

Choosing Parameters for PSO

Time Comparisons

Conclusión

“Our experiment shows that PSO is a potential candidate for training neural network. Further experimentation could help us to make our claim strong.”

Future Work

Memetic Approach
and
Lamarckian Approach

Expand Word Corpus
using PubMed
dataset

Different Inertia
Scheme

Use Parallel
thread-PSO

Questions?