Algorithm 1 Differential Evolution

Generate initial population of size *pplSize*
Calculate fitness of each individual in population
for each iteration till *maxiter* **do**
    for every individual **do**
        Set individual as *parent*
        Set or generate the factors β* and CR*
        Select a random/the best *targetvector**
        Select randomly individuals for *difference vector*
        Do **Mutation*** to get *unitvector*
        Crossover* the *parent* and the *unitvector* to get a *child*
        Do **Selection** over the *parent* and the *child* based on their *fitness*
    end for
    Save statistics of the *generation*
end for
* Differences of DE variants

Algorithm 2 Fitness function

Define the *Neural Network*
Set *Adam* for optimizer
Set *Binary Cross Entropy with Sigmoid layer* as loss function
for every epoch **do**
    Shuffle indexes
    for each training batch **do**
        Set gradients to zero
        Train the *Neural Network* on the batch of Data to get *output*
        Calculate the **training loss** for *output* vs *labels*
    end for
    for each validation batch **do**
        Validate the *Neural Network* on the batch of Data to get *output*
        Calculate the **validation loss** for *output* vs *labels*
    end for
    if Validation loss $<$ Current Best **then**
        Set Current Best = Validation Loss
    else
        Count Stop
    end if
    if Stop > 3 **then**
        Return Current Best, Neural Network Config and id
    end if
end for