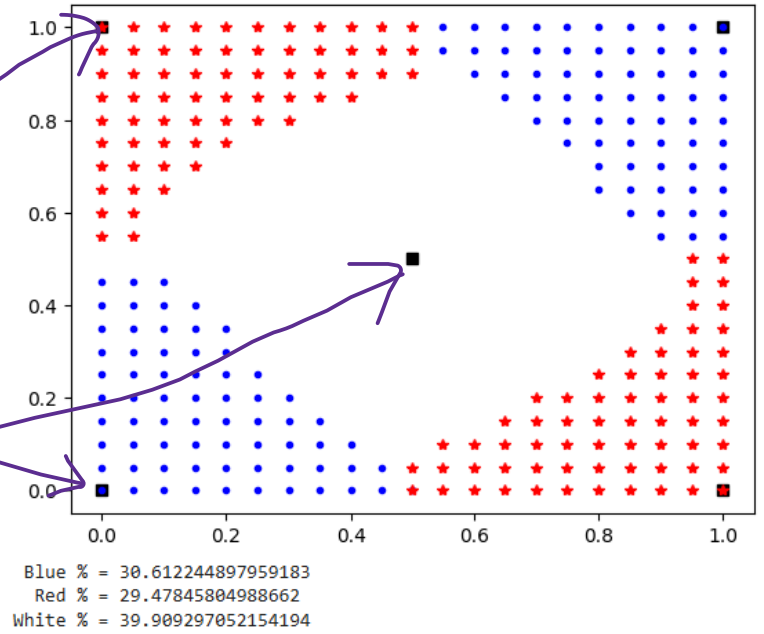
HW4 (HPO): Finding Optimized DNNs using ES(1+1)1/5 for XOR with 3 classes

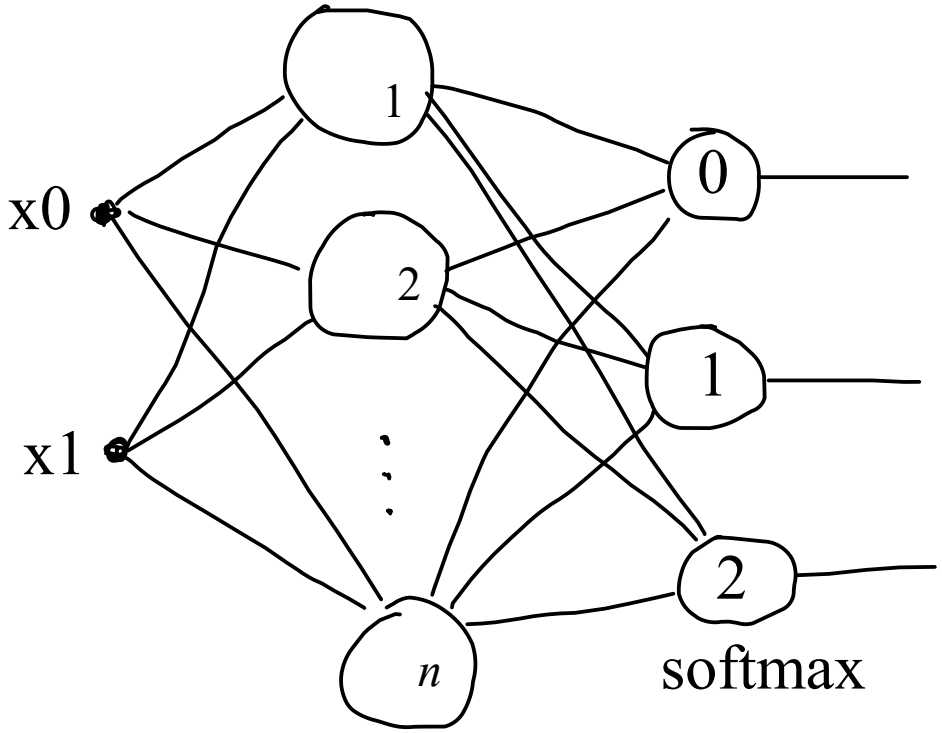
Important for your class research projects

x0	x1	XOR3
0.0	0.0	0
0.0	1.0	1
1.0	0.0	1
1.0	1.0	0
0.5	0.5	2



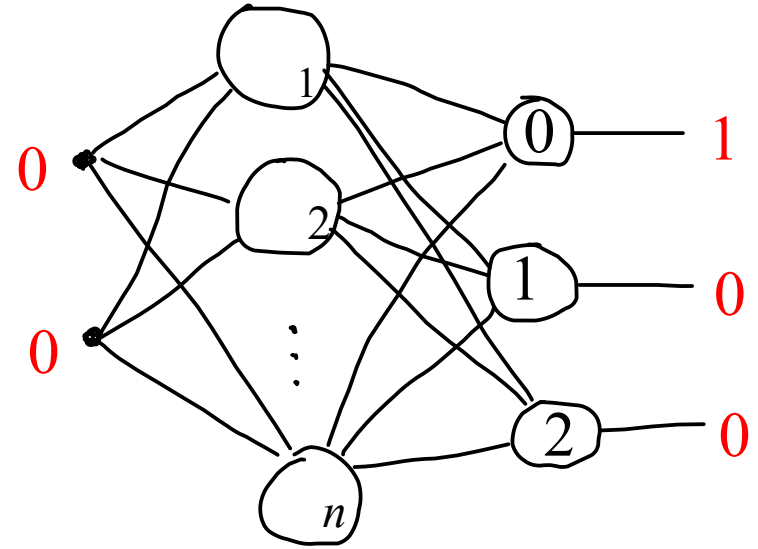
LTU CS CJ Chung

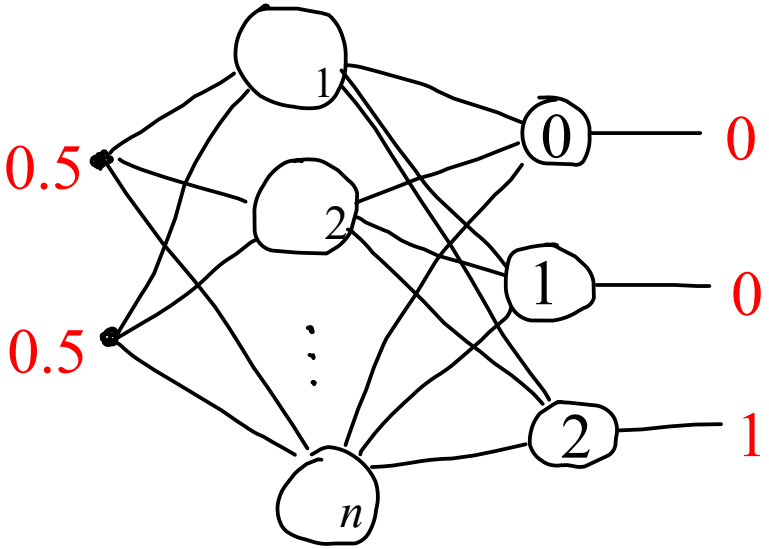
Model



n: ?

Act_fn: ?





HW4: Requirements (1 /2)

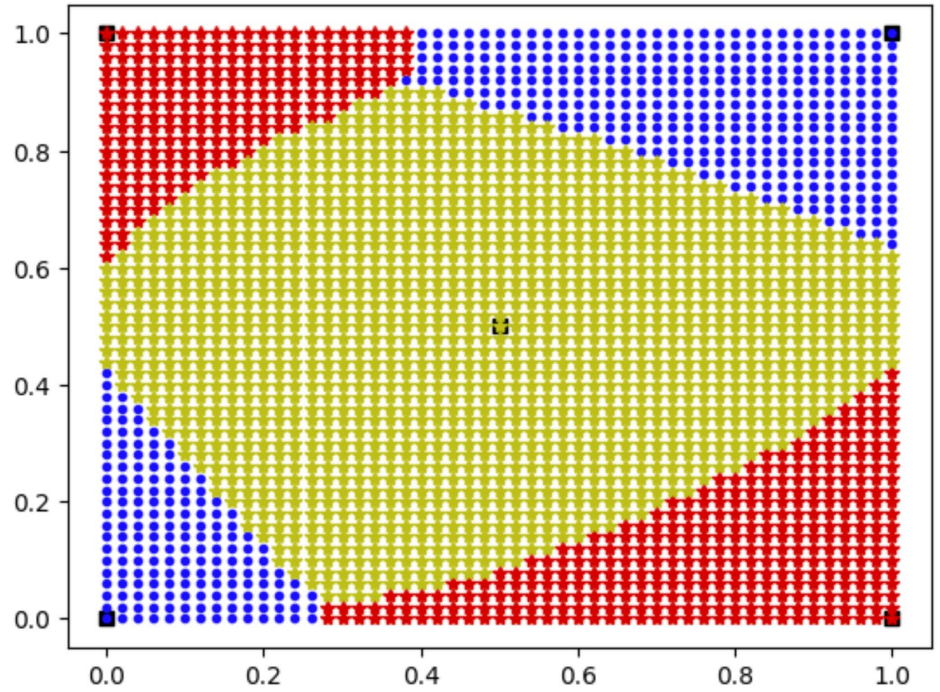
- ES1115_HPO_XOR.ipynb file was provided on Canvas (on Sep 29)
- XOR3.ipynb file is provided on Canvas
- 6 Hyper Parameters:
  - # of hidden neurons in one hidden layer
  - Learning rate
  - Batch size
  - # of epochs
  - 5 activation functions for hidden neurons: ('relu', 'elu', 'sigmoid', 'tanh', 'leaky_relu')
  - 3 optimizer options: ('SGD', 'RMSprop', 'Adam')

Do not use loss_func as a hyper param. For this multiclass classification problem, use 'sparse_categorical_crossentropy', since target labels are integers

HW4: Requirements (2 /2)

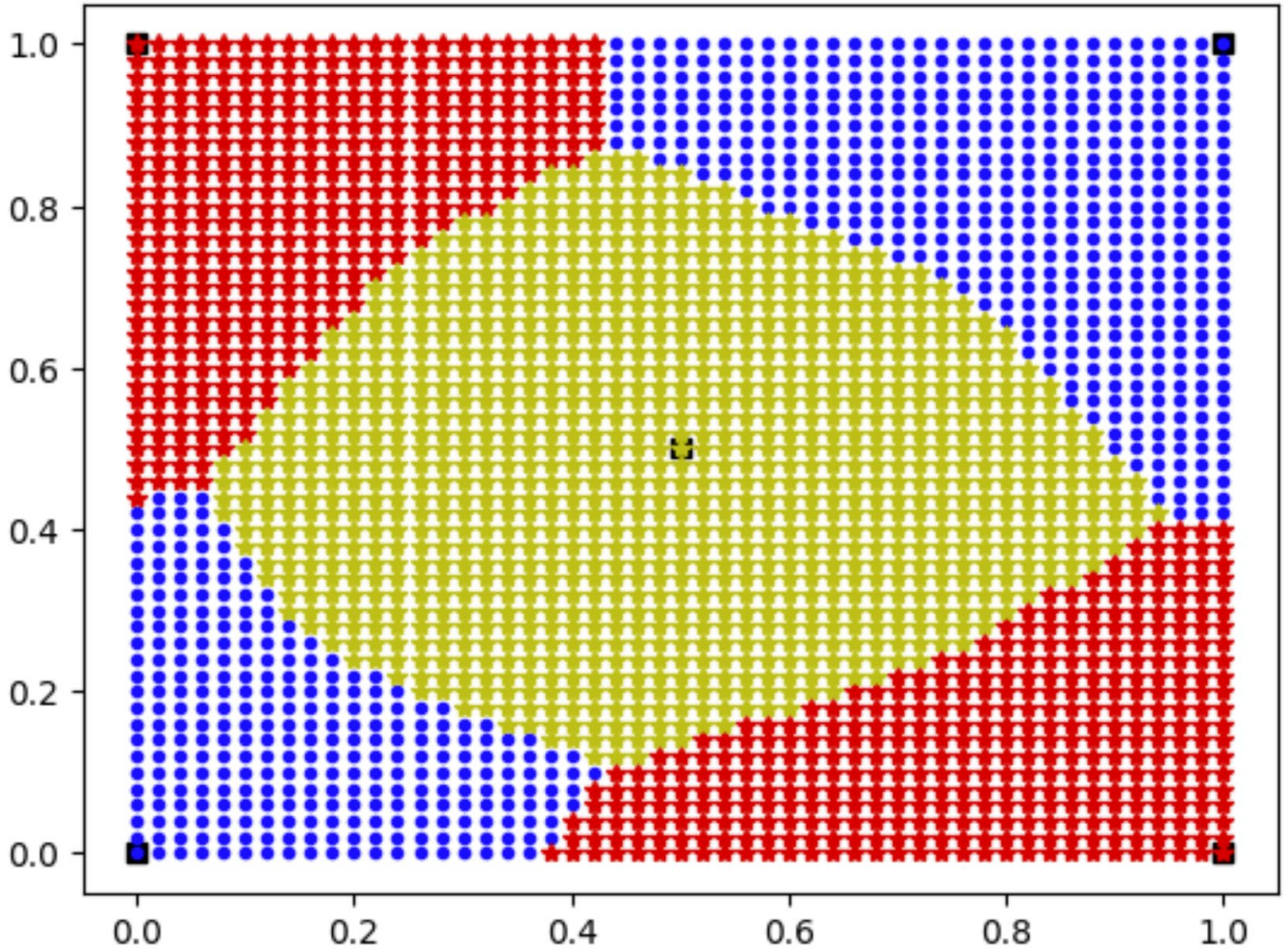
- Run 3 times (Trial = 3)
- If model loss is less than 0.01,
  - print something like:
  - and plot 2D mesh grid diagram and print percentages shown right
- Note that the decision boundary plot must be displayed whenever acceptable solution is found

```
************************* Trial # = 1
Acceptable solution found after 22 generations:
#neurons=4, lr=0.400, bsize=1, epochs=300, actF=elu, optim=Adam
Loss_Eval=0.00027059036074206233
```



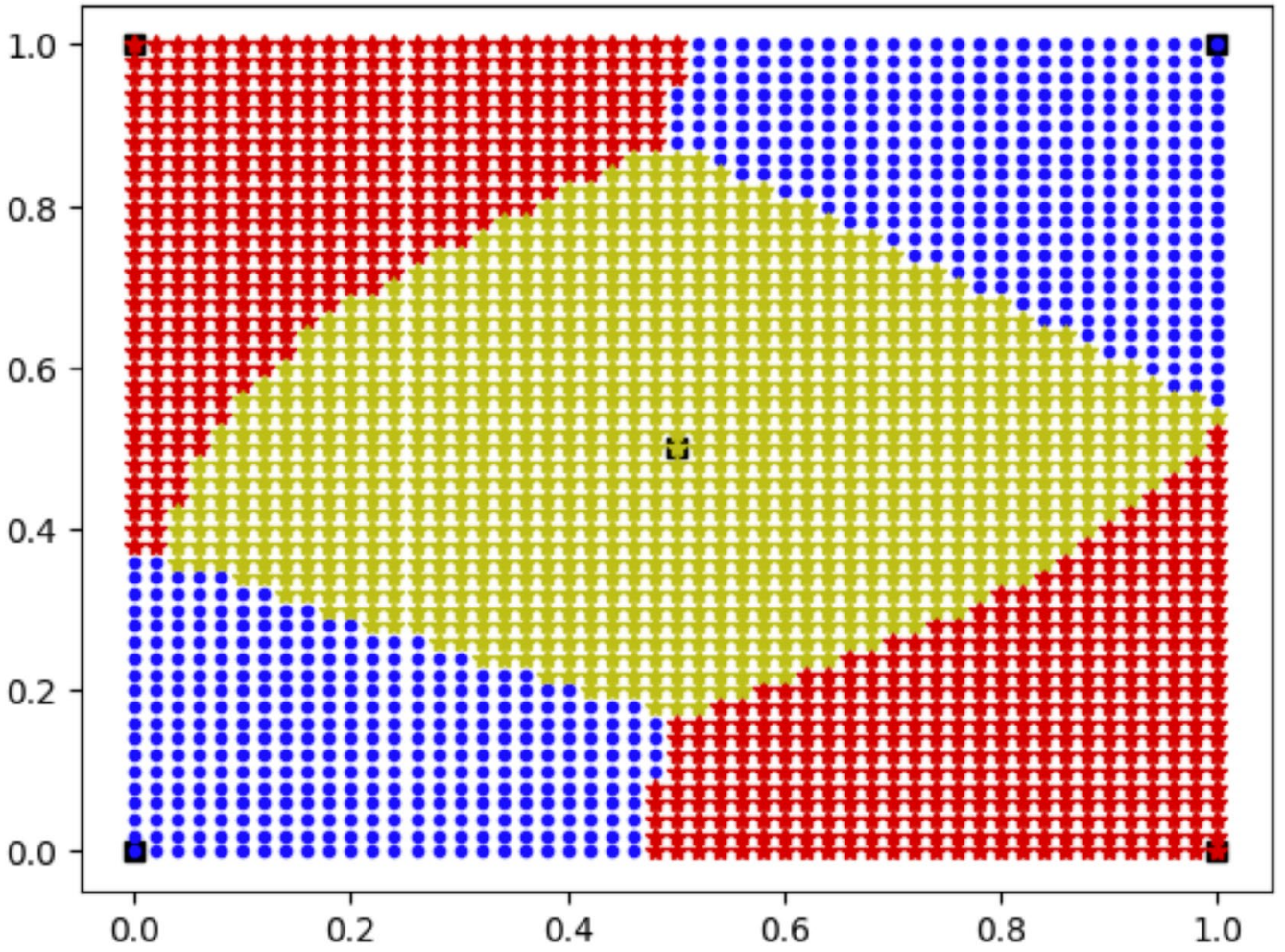

```
Red % = 21.91464821222607
Blue % = 20.26143790849673
Yellow % = 57.8239138792772
```

```
************************* Trial # = 2
Acceptable solution found after 3 generations:
#neurons=12, lr=0.010, bsize=1, epochs=300, actF=relu, optim=RMSprop
Loss_Eval=0.0017887648427858949
```




```
Red % = 29.64244521337947
Blue % = 29.757785467128027
Yellow % = 40.5997693194925
```

```
************************* Trial # = 3
Acceptable solution found after 4 generations:
#neurons=13, lr=0.010, bsize=5, epochs=300, actF=relu, optim=RMSprop
Loss_Eval=0.004184751305729151
```




```
Red % = 32.17993079584775
Blue % = 27.989234909650136
Yellow % = 39.83083429450212
```

HW4: Range or Options of Hyper Parameters

0	1	2	3	4	5
# neurons	Learning rate	Batch size	# epochs	Activation func	Optimizer
2 ~ 16	0.01 ~ 1.5	1 ~ 5	100 ~ 500		

```
optimz = ('SGD', 'RMSprop', 'Adam')
```

```
act_func = ('relu', 'elu', 'sigmoid', 'tanh', 'leaky_relu')
```

for hidden neurons

Note that the params of ES1115_HPO_XOR.ipynb *are different.*

To scale the step size relative to the range of a parameter

1. **Define the range of the parameter**: Let's say the parameter x you are optimizing has a range $[x_{\min}, x_{\max}]$.
2. **Relative step size**: You want the step size σ to be meaningful in the context of the parameter's range. A simple way to do this is by expressing σ as a fraction of the parameter range:

$$\sigma_{\text{scaled}} = \sigma \times (x_{\max} - x_{\min})$$

   This ensures that the step size remains proportional to the parameter's allowed values.
3. **Update rule (1/5th success rule)**: After generating a new candidate solution, observe whether the mutation was successful (i.e., if it leads to a better solution). If the success rate is higher than 1/5, increase the step size; otherwise, decrease it:
   - If the success rate $s > \frac{1}{5}$, increase σ_{scaled} by a constant factor (e.g., 1.5).
   - If $s < \frac{1}{5}$, decrease σ_{scaled} by a constant factor (e.g., 0.85).

   Mathematically:

$$\sigma_{\text{scaled}} = \begin{cases} \sigma_{\text{scaled}} \times c_{\text{up}} & \text{if success rate} > \frac{1}{5} \\ \sigma_{\text{scaled}} \times c_{\text{down}} & \text{if success rate} < \frac{1}{5} \end{cases}$$

   where c_{up} and c_{down} are typically around 1.5 and 0.85, respectively

HW4 Checklist

- Course disclaimer including your name
- Algorithm for HPO with 6 parameters is correct and works well?
- Must use “scaled” stepsize. See slide 8
- Must define **es1115()** function called in the “Trial” loop
- Must define a function to plot decision boundary graph
- Your submitted .ipynb file includes the run result of all the acceptable solutions with
  - Acceptable param list displayed together with loss value
  - A correct mesh decision boundary graph

Notes

- Warning: start this assignment early. It will take at least 10+ minutes to run 3 trials
- restarting runtime might be needed. (Colab)
- “Retest” your code before submission
- Make sure to include run results before submission