Powerplay

Evolutionary powerplay for Nascence materials

- This example contains a search for classifiers

Jan Koutnik, IDSIA, E−mail: hkou.idsia@ch

Imports

In[9]:=
```
Import["../libMLP/bp9.m"]
```

In[26]:=
```
Import["../libNES/nes10.m"]
```

In[209]:=
```
Import["../libCoSyNE/libCosyne9.m"]
```

VM

Vitual material is a simple MLP here :

In[407]:=
```
actF := 2 / (1 + Exp[-#]) - 1 &
```

In[410]:=
```
actF = Tanh;
```

In[656]:=
```
vm = randomNet[{10, 50, 20, 1}];
```

In[463]:=
```
evalVM[vm_, x_] := Last@fwdPass[vm, x]
```

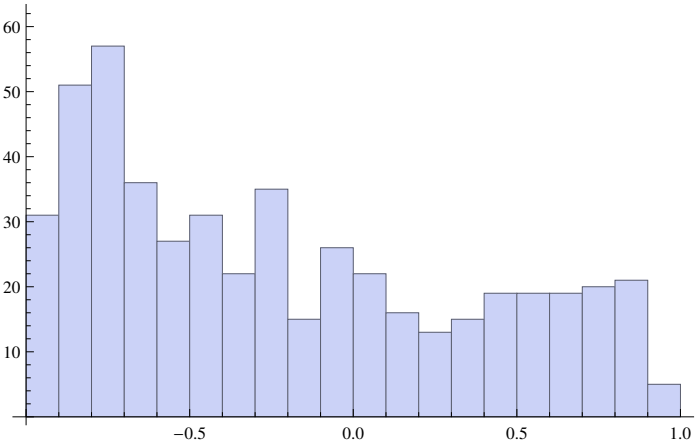
In[657]:=
```
res = Sort@Flatten[evalVM[vm, #] & /@ RandomReal[{-1, 1}, {500, 10}]];
Min@res
Max@res
```

Out[658]= −0.972087

Out[659]= 0.957285

In[660]:=
```
Histogram[res, 20]
```

Out[660]=



In[416]:=
```
evalVM[vm_, config_, input_] := evalVM[vm, Join[config, input]]
```

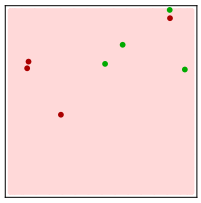
In[643]:=
```
plotBounds[vm_, config_, set_] :=
 With[{points = (First /@ #) & /@ SortBy[GatherBy[set, Last], #[[1, 2, 1]] &]}, Show[{
    ContourPlot[evalVM[vm, config, {x, y}], {x, -1, 1},
     {y, -1, 1}, Contours → {0}, ContourShading → {LightRed, LightGreen},
     ContourStyle → Directive[Black, Dashed], FrameTicks → False, ImageSize → 100],
    ListPlot[points, PlotStyle → {Darker@Red, Darker@Green}]
   }]]
```

In[661]:=
```
evalVM[vm, config, {1, 1}]
```

Out[661]=
```
{-0.601582}
```

In[662]:=
```
plotBounds[vm, config, trainingSet]
```

Out[662]=



Powerplay Example - 2-space Classifiers

In[8]:=
```
SetDirectory[NotebookDirectory[]]
```

Out[8]=
```
/home/koutnij/Dropbox/math/nascence
```

In this example a space of classifiers is explored.

In[1]:=
```
maxTasks = 1000;
```

In[814]:=
```
SeedRandom[30];
taskSet = RandomReal[{-1, 1}, {maxTasks, 2}];
```

In[480]:=
```
currentTask = 1;
```

First point in the task set :

In[816]:=
```
trainingSet = {{taskSet[[1]], {1}}}
```

Out[816]=
```
{{{-0.71618, 0.607906}, {1}}}
```

In[698]:=
```
trainingSet =
  {{taskSet[[1]], {1}}, {taskSet[[2]], {-1}}, {taskSet[[3]], {1}}, {taskSet[[4]], {-1}},
   {taskSet[[5]], {1}}, {taskSet[[6]], {-1}}, {taskSet[[7]], {1}}, {taskSet[[8]], {-1}}};
```

XOR :

In[682]:=
```
xorSet = MapThread[Append[{#1}, {#2}] &, {Tuples[{-.9, .9}, 2], {-1, 1, 1, -1}}]
```

Out[682]=
```
{{{-0.9, -0.9}, {-1}}, {{-0.9, 0.9}, {1}}, {{0.9, -0.9}, {1}}, {{0.9, 0.9}, {-1}}}
```

In[684]:=
```
trainingSet = xorSet;
```

In[789]:=
```
fitFn[g_, set_] := With[
  {diff = Flatten[(evalVM[vm, g, #] & /@ set[[All, 1]]) - set[[All, 2]]]}, Total[diff^2]]
```

In[790]:=
```
fitFn[g_] := fitFn[g, trainingSet]
```

In[788]:=
```
solvesAllTasksQ[g_, set_] :=
 Equal[Sign@Flatten[(evalVM[vm, g, #] & /@ set[[All, 1]])], Flatten@set[[All, 2]]]
```

In[706]:=
```
solvesAllTasksQ[g_] := solvesAllTasksQ[g, trainingSet]
```

In[791]:=
```
optimize[pop_, fitFn_, trainingSet_, nGen_] :=
 Module[{popTmp}, NestWhile[(popTmp = coSyNEstep[#, fitFn[#, trainingSet] &,
        minimize → True, mutate → 0.8, permuteAll → True, verbose → False, elite → 1];
      Print[{popTmp[[1, 1]], solvesAllTasksQ[popTmp[[1, 2]], trainingSet]}];
      Print@plotBounds[vm, popTmp[[1, 2]], trainingSet]; popTmp) &,
   pop, Not[solvesAllTasksQ[#[[1, 2]], trainingSet]] &, 1,
   nGen]]
```

In[792]:=
```
reevaluate[pop_, fitFn_] := SortBy[{fitFn[#], #} & /@ pop[[All, 2]], First]
```

In[793]:=
```
appendTask[set_, config_] := With[{point = taskSet[[Length[set] + 1]]},
  Append[set, {point, -Sign[evalVM[vm, config, point]]}]]
```

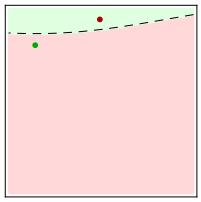
In[807]:=
```
powerplay[fitFn_, popSize_, nGen_, nTasks_] := Module[{pop, set},
  set = {{taskSet[[1]], {1}}}; (*first task*)
  pop = SortBy[
    newRandomPop[popSize, dim, NormalDistribution[0, 1], fitFn[#, set] &], First];
  Print["Generated population"];
  Nest[
   (Print["# of tasks : " <> ToString@Length[set]];
    pop = optimize[pop, fitFn, set, nGen]; (*optimize*)
    set = appendTask[set, pop[[1, 2]]]; (*add next task*)
    pop = reevaluate[pop, fitFn[#, set] &];
    pop) &
   , pop, nTasks]
 ]
```

Example Experiment

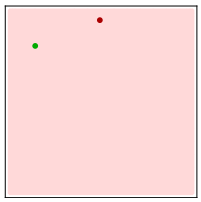
Powerplay with population size of 8, 30 generations of CoSyNE and 10 consecutive classification tasks to solve :

In[818]:=
```
powerplay[fitFn, 8, 30, 10]
```

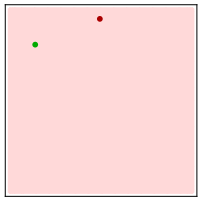
```
Generated population

# of tasks : 1

# of tasks : 2

{2.34823, False}
```



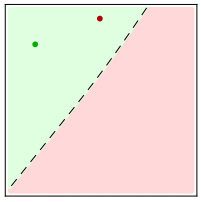
```
{2.09571, False}
```



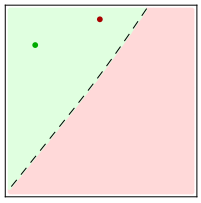
```
{2.00215, False}
```



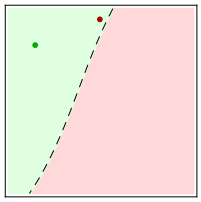
```
{1.91676, False}
```



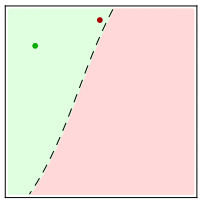
```
{1.91676, False}
```



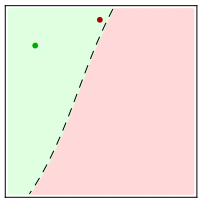
```
{1.69548, False}
```



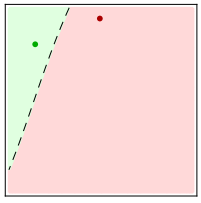
```
{1.69548, False}
```



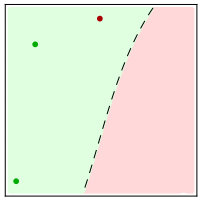
```
{1.69548, False}
```



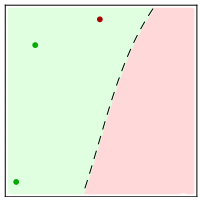
```
{1.5368, True}
```



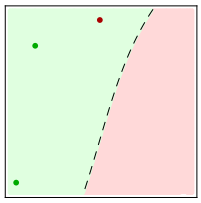
```
# of tasks : 3

{2.30423, False}
```



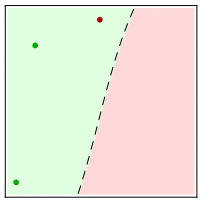
```
{2.30423, False}
```



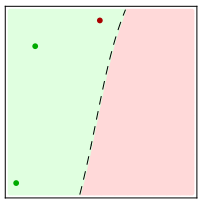
```
{2.30423, False}
```



{2.28959, False}



{2.28359, False}



{2.23056, False}