

Compression Normal-Absolute Poker Dataset

March 25, 2017

Compute the performance of MAB methods

```
In [17]: import numpy as np
import time
import sys
import matplotlib.pyplot as plt
from sklearn import metrics
%matplotlib inline
plt.rcParams['figure.figsize'] = (15, 6)
```

0.1 Load BOKEH library

```
In [18]: from bokeh.layouts import row, gridplot
from bokeh.plotting import figure, output_notebook, show
from bokeh.models import Legend
TOOLS = 'box_zoom,box_select,crosshair,resize,reset,lasso_select,pan,save,poly_select,tap'
output_notebook()
```

1 Compare the accuracy of the models

1.1 Load the pruned algorithm from normal prune

```
In [19]: ucb1 = np.load('/Users/saleem/Desktop/banditsbook/python_poker/UCB1/AccuracyAfterPrune.npy')
EpsilonGreedy = np.load('/Users/saleem/Desktop/banditsbook/python_poker/EpsilonGreedy/AccuracyAfterPrune.npy')
AnnealingEpsilonGreedy = np.load('/Users/saleem/Desktop/banditsbook/python_poker/AnnealingEpsilonGreedy/AccuracyAfterPrune.npy')
Softmax = np.load('/Users/saleem/Desktop/banditsbook/python_poker/Softmax/AccuracyAfterPrune.npy')
AnnealingSoftmax = np.load('/Users/saleem/Desktop/banditsbook/python_poker/AnnealingSoftmax/AccuracyAfterPrune.npy')
Exp3 = np.load('/Users/saleem/Desktop/banditsbook/python_poker/Exp3/AccuracyAfterPrune.npy')
Hedge = np.load('/Users/saleem/Desktop/banditsbook/python_poker/Hedge/AccuracyAfterPrune.npy')
ThompsonSampling = np.load('/Users/saleem/Desktop/banditsbook/python_poker/ThompsonSampling/AccuracyAfterPrune.npy')
Accuracy = np.load('/Users/saleem/Desktop/banditsbook/python_poker/AccuracyBeforePrune.npy')
```

1.2 Load the pruned algorithm from absolute prune

```
In [20]: ucb1_absolute = np.load('/Users/saleem/Desktop/banditsbook_moving_output/python_poker/UCB1/AccuracyAfterPrune.npy')
EpsilonGreedy_absolute = np.load('/Users/saleem/Desktop/banditsbook_moving_output/python_poker/EpsilonGreedy/AccuracyAfterPrune.npy')
AnnealingEpsilonGreedy_absolute = np.load('/Users/saleem/Desktop/banditsbook_moving_output/python_poker/AnnealingEpsilonGreedy/AccuracyAfterPrune.npy')
```

```

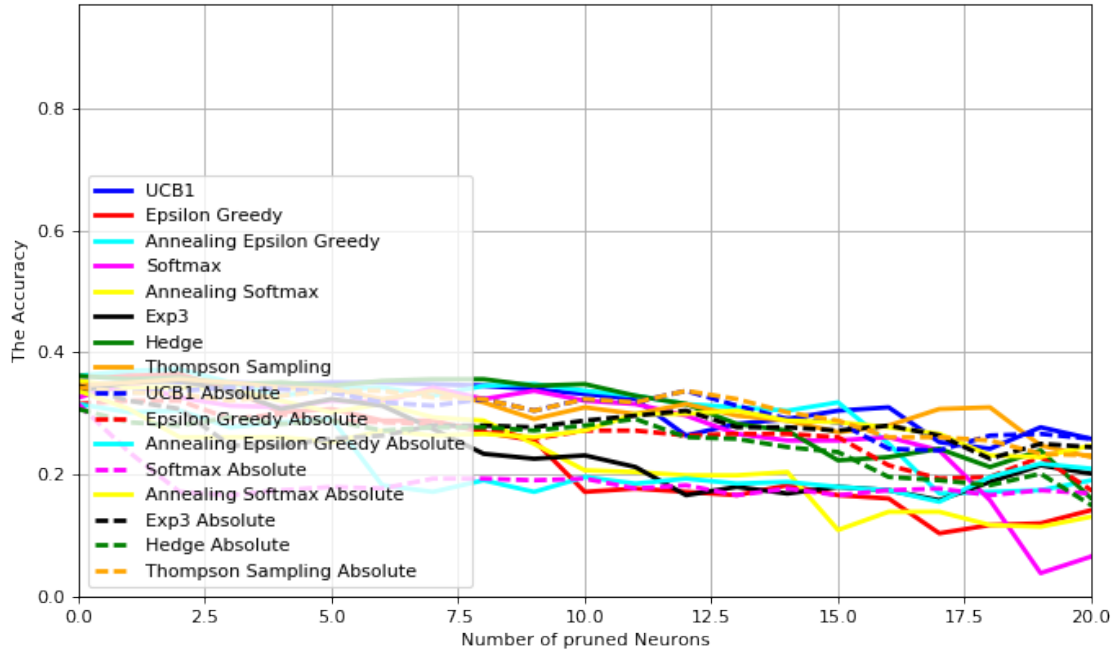
Softmax_absulate = np.load('/Users/saleemameen/Desktop/banditsbook_moving_output/python_
AnnealingSoftmax_absulate = np.load('/Users/saleemameen/Desktop/banditsbook_moving_outpu
Exp3_absulate = np.load('/Users/saleemameen/Desktop/banditsbook_moving_output/python_pok
Hedge_absulate = np.load('/Users/saleemameen/Desktop/banditsbook_moving_output/python_po
ThompsonSampling_absulate = np.load('/Users/saleemameen/Desktop/banditsbook_moving_outpu

```

```

In [21]: fig = plt.figure(figsize=(10, 6), dpi=80)
ax = fig.add_subplot(111)
N = len(ucb1)
## necessary variables
ind = np.arange(N) # the x locations for the groups
### Normal algoritms
plt.plot(ind , ucb1 , color="blue", linewidth=2.5, linestyle="-", label="UCB1")
plt.plot(ind , EpsilonGreedy, color="red", linewidth=2.5, linestyle="-", label="EpsilonGreedy")
plt.plot(ind , AnnealingEpsilonGreedy, color="cyan", linewidth=2.5, linestyle="-", label="AnnealingEpsilonGreedy")
plt.plot(ind , Softmax, color="magenta", linewidth=2.5, linestyle="-", label="Softmax")
plt.plot(ind , AnnealingSoftmax, color="yellow", linewidth=2.5, linestyle="-", label="AnnealingSoftmax")
plt.plot(ind , Exp3, color="black", linewidth=2.5, linestyle="-", label="Exp3")
plt.plot(ind , Hedge, color="green", linewidth=2.5, linestyle="-", label="Hedge")
plt.plot(ind , ThompsonSampling, color="orange", linewidth=2.5, linestyle="-", label="ThompsonSampling")
### Absuluate once algoritms
plt.plot(ind , ucb1_absulate , color="blue", linewidth=2.5, linestyle="--", label="UCB1_absulate")
plt.plot(ind , EpsilonGreedy_absulate, color="red", linewidth=2.5, linestyle="--", label="EpsilonGreedy_absulate")
plt.plot(ind , AnnealingEpsilonGreedy_absulate, color="cyan", linewidth=2.5, linestyle="--", label="AnnealingEpsilonGreedy_absulate")
plt.plot(ind , Softmax_absulate, color="magenta", linewidth=2.5, linestyle="--", label="Softmax_absulate")
plt.plot(ind , AnnealingSoftmax_absulate, color="yellow", linewidth=2.5, linestyle="--", label="AnnealingSoftmax_absulate")
plt.plot(ind , Exp3_absulate, color="black", linewidth=2.5, linestyle="--", label="Exp3_absulate")
plt.plot(ind , Hedge_absulate, color="green", linewidth=2.5, linestyle="--", label="Hedge_absulate")
plt.plot(ind , ThompsonSampling_absulate, color="orange", linewidth=2.5, linestyle="--", label="ThompsonSampling_absulate")
#####
plt.legend(loc = 3)
plt.axis([0, 20, 0, 0.97])
plt.xlabel('Number of pruned Neurons')
plt.ylabel('The Accuracy')
plt.grid(True)
plt.show()

```



```
In [22]: p1 = figure(title="The Performance over the number of neurons' pruned", tools=TOOLS)
# Normal pruned
#p1.circle(ind, ucb1, legend="ucb1", color="orange")
p1.line(ind, ucb1, legend="ucb1", line_color="orange", line_width=2)
#p1.square(ind, EpsilonGreedy, legend="Epsilon Greedy", fill_color=None, line_color="red")
p1.line(ind, EpsilonGreedy, legend="Epsilon Greedy", line_color="red", line_width=2)
#p1.ellipse(ind, AnnealingEpsilonGreedy, legend="Annealing Epsilon Greedy", line_color="blue")
p1.line(ind, AnnealingEpsilonGreedy, legend="Annealing Epsilon Greedy", line_color="blue", line_width=2)
#p1.diamond(ind, Softmax, legend="Softmax", line_color="green")
p1.line(ind, Softmax, legend="Softmax", line_color="green", line_width=2)
#p1.arc(ind, AnnealingSoftmax, legend="Annealing Softmax", line_color="grey", end_angle=0.5)
p1.line(ind, AnnealingSoftmax, legend="Annealing Softmax", line_color="grey", line_width=2)
#p1.oval(ind, Exp3, legend="Exp3", line_color="black", height=0.01, width=0.01)
p1.line(ind, Exp3, legend="Exp3", line_color="black", line_width=2)
#p1.arc(ind, Hedge, legend="Hedge", line_color="yellow")
#p1.triangle(ind, Hedge, legend="Hedge", line_color="yellow")
p1.line(ind, Hedge, legend="Hedge", line_color="yellow", line_width=2)
#p1.square_cross(ind, ThompsonSampling, legend="Thompson Sampling", line_color="pink")
p1.line(ind, ThompsonSampling, legend="Thompson Sampling", line_color="pink", line_width=2)
#p1.line(ind, Exp3, legend="2*sin(x)", line_dash=(4, 4), line_color="orange", line_width=2)
#p1.square(ind, Hedge, legend="3*sin(x)", fill_color=None, line_color="brown")
##### =====
# Absulate pruned
p1.circle(ind, ucb1_absulate, legend="ucb1 Absolute", line_color="orange", line_width=2)
p1.line(ind, ucb1_absulate, legend="ucb1 Absolute", line_color="orange", line_width=2)
```

```

#p1.square(ind, EpsilonGreedy, legend="Epsilon Greedy", fill_color=None, line_color="red")
p1.circle(ind, EpsilonGreedy_absulate, legend="Epsilon Greedy Absolute", line_color="red")
p1.line(ind, EpsilonGreedy_absulate, legend="Epsilon Greedy Absolute", line_color="red")
#p1.ellipse(ind, AnnealingEpsilonGreedy, legend="Annealing Epsilon Greedy", line_color="red")
p1.circle(ind, AnnealingEpsilonGreedy_absulate, legend="Annealing Epsilon Greedy Absolute")
p1.line(ind, AnnealingEpsilonGreedy_absulate, legend="Annealing Epsilon Greedy Absolute")
#p1.diamond(ind, Softmax, legend="Softmax", line_color="green")
p1.circle(ind, Softmax_absulate, legend="Softmax Absolute", line_color="green", line_width=2)
p1.line(ind, Softmax_absulate, legend="Softmax Absolute", line_color="green", line_width=2)
#p1.arc(ind, AnnealingSoftmax, legend="Annealing Softmax", line_color="grey", end_angle=90)
p1.circle(ind, AnnealingSoftmax_absulate, legend="Annealing Softmax Absolute", line_color="grey")
p1.line(ind, AnnealingSoftmax_absulate, legend="Annealing Softmax Absolute", line_color="grey")
#p1.oval(ind, Exp3, legend="Exp3", line_color="black", height=0.01, width=0.01)
p1.circle(ind, Exp3_absulate, legend="Exp3 Absolute", line_color="black", line_width=2)
p1.line(ind, Exp3_absulate, legend="Exp3 Absolute", line_color="black", line_width=2)
#p1.arc(ind, Hedge, legend="Hedge", line_color="yellow")
#p1.triangle(ind, Hedge, legend="Hedge", line_color="yellow")
p1.circle(ind, Hedge_absulate, legend="Hedge Absolute", line_color="yellow", line_width=2)
p1.line(ind, Hedge_absulate, legend="Hedge Absolute", line_color="yellow", line_width=2)
#p1.square_cross(ind, ThompsonSampling, legend="Thompson Sampling", line_color="pink")
p1.circle(ind, ThompsonSampling_absulate, legend="Thompson Sampling Absolute", line_color="pink")
p1.line(ind, ThompsonSampling_absulate, legend="Thompson Sampling Absolute", line_color="pink")
#p1.line(ind, Exp3, legend="2*sin(x)", line_dash=(4, 4), line_color="orange", line_width=2)
#p1.square(ind, Hedge, legend="3*sin(x)", fill_color=None, line_color="brown")
p1.title.align = "center"
show(p1)
#show(gridplot(p1, p2, ncols=2, plot_width=400, plot_height=400)) # open a browser

```

1.3 Comparing All algorithms with the model before pruning

```

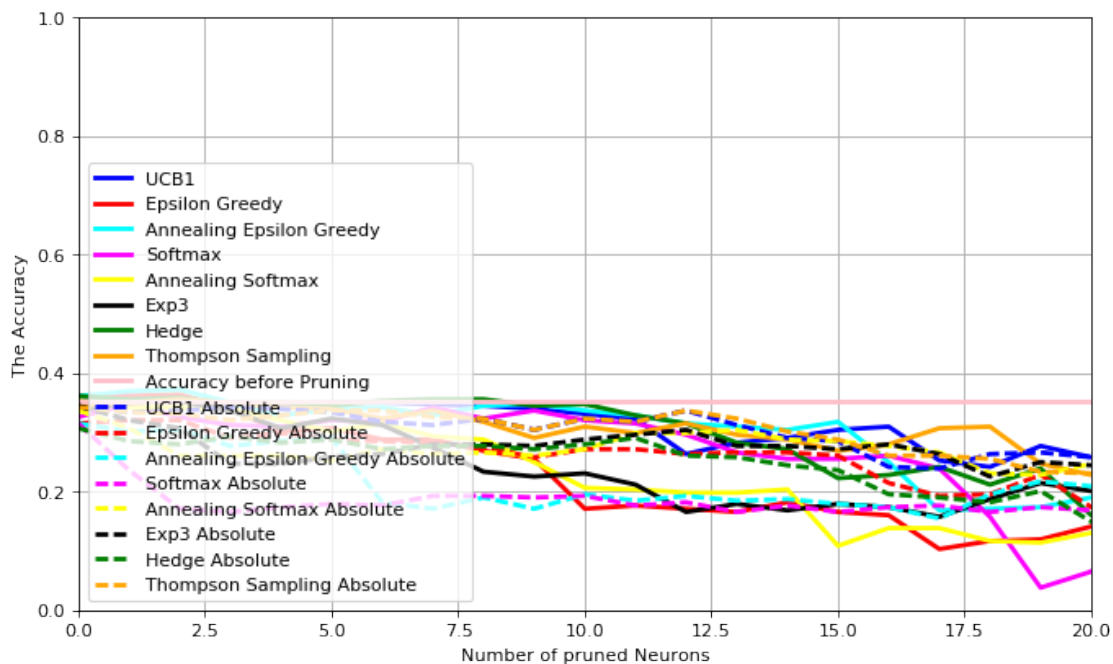
In [23]: fig = plt.figure(figsize=(10, 6), dpi=80)
ax = fig.add_subplot(111)
N = len(ucb1)
Acc = [Accuracy for col in range(N)]
## necessary variables
ind = np.arange(N) # the x locations for the groups
plt.plot(ind, ucb1, color="blue", linewidth=2.5, linestyle="-", label="UCB1")
plt.plot(ind, EpsilonGreedy, color="red", linewidth=2.5, linestyle="-", label="Epsilon Greedy")
plt.plot(ind, AnnealingEpsilonGreedy, color="cyan", linewidth=2.5, linestyle="-", label="Annealing Epsilon Greedy")
plt.plot(ind, Softmax, color="magenta", linewidth=2.5, linestyle="-", label="Softmax")
plt.plot(ind, AnnealingSoftmax, color="yellow", linewidth=2.5, linestyle="-", label="Annealing Softmax")
plt.plot(ind, Exp3, color="black", linewidth=2.5, linestyle="-", label="Exp3")
plt.plot(ind, Hedge, color="green", linewidth=2.5, linestyle="-", label="Hedge")
plt.plot(ind, ThompsonSampling, color="orange", linewidth=2.5, linestyle="-", label="Thompson Sampling")
plt.plot(ind, Acc, color="pink", linewidth=2.5, linestyle="-", label="Accuracy before pruning")
### Absuluate once algoritms
plt.plot(ind, ucb1_absulate, color="blue", linewidth=2.5, linestyle="--", label="UCB1 Absolute")
plt.plot(ind, EpsilonGreedy_absulate, color="red", linewidth=2.5, linestyle="--", label="Epsilon Greedy Absolute")

```

```

plt.plot(ind , AnnealingEpsilonGreedy_absulate, color="cyan", linewidth=2.5, linestyle=
plt.plot(ind , Softmax_absulate, color="magenta", linewidth=2.5, linestyle="--", label=
plt.plot(ind , AnnealingSoftmax_absulate, color="yellow", linewidth=2.5, linestyle="--"
plt.plot(ind , Exp3_absulate, color="black", linewidth=2.5, linestyle="--", label="Exp3
plt.plot(ind , Hedge_absulate, color="green", linewidth=2.5, linestyle="--", label="Hed
plt.plot(ind , ThompsonSampling_absulate, color="orange", linewidth=2.5, linestyle="--"
plt.legend(loc = 3)
plt.axis([0, 20, 0, 1])
plt.xlabel('Number of pruned Neurons')
plt.ylabel('The Accuracy')
plt.grid(True)
plt.show()

```



```

In [24]: p1 = figure(title="The Performance over the number of neurons' pruned", tools=TOOLS)
# Normal pruned
#p1.circle(ind, ucb1, legend="ucb1", color="orange")
p1.line(ind, ucb1, legend="ucb1", line_color="orange", line_width=2)
#p1.square(ind, EpsilonGreedy, legend="Epsilon Greedy", fill_color=None, line_color="red")
p1.line(ind, EpsilonGreedy, legend="Epsilon Greedy", line_color="red", line_width=2)
#p1.ellipse(ind, AnnealingEpsilonGreedy, legend="Annealing Epsilon Greedy", line_color="blue")
p1.line(ind, AnnealingEpsilonGreedy, legend="Annealing Epsilon Greedy", line_color="blue", line_width=2)
#p1.diamond(ind, Softmax, legend="Softmax", line_color="green")
p1.line(ind, Softmax, legend="Softmax", line_color="green", line_width=2)
#p1.arc(ind, AnnealingSoftmax, legend="Annealing Softmax", line_color="grey", end_angle=90)
p1.line(ind, AnnealingSoftmax, legend="Annealing Softmax", line_color="grey", line_width=2)

```

```

#p1.oval(ind, Exp3, legend="Exp3", line_color="black", height=0.01, width=0.01)
p1.line(ind, Exp3, legend="Exp3", line_color="black", line_width=2)
#p1.arc(ind, Hedge, legend="Hedge", line_color="yellow")
#p1.triangle(ind, Hedge, legend="Hedge", line_color="yellow")
p1.line(ind, Hedge, legend="Hedge", line_color="yellow", line_width=2)
#p1.square_cross(ind, ThompsonSampling, legend="Thompson Sampling", line_color="pink")
p1.line(ind, ThompsonSampling, legend="Thompson Sampling", line_color="pink", line_width=2)
#p1.line(ind, Exp3, legend="2*sin(x)", line_dash=(4, 4), line_color="orange", line_width=2)
#p1.square(ind, Hedge, legend="3*sin(x)", fill_color=None, line_color="brown")
##### =====
# Absulate pruned
# Absulate pruned
p1.circle(ind, ucb1_absulate, legend="ucb1 Absolute", line_color="orange", line_width=2)
p1.line(ind, ucb1_absulate, legend="ucb1 Absolute", line_color="orange", line_width=2)
#p1.square(ind, EpsilonGreedy, legend="Epsilon Greedy", fill_color=None, line_color="red")
p1.circle(ind, EpsilonGreedy_absulate, legend="Epsilon Greedy Absolute", line_color="red")
p1.line(ind, EpsilonGreedy_absulate, legend="Epsilon Greedy Absolute", line_color="red")
#p1.ellipse(ind, AnnealingEpsilonGreedy, legend="Annealing Epsilon Greedy", line_color="red")
p1.circle(ind, AnnealingEpsilonGreedy_absulate, legend="Annealing Epsilon Greedy Absolute", line_color="red")
p1.line(ind, AnnealingEpsilonGreedy_absulate, legend="Annealing Epsilon Greedy Absolute", line_color="red")
#p1.diamond(ind, Softmax, legend="Softmax", line_color="green")
p1.circle(ind, Softmax_absulate, legend="Softmax Absolute", line_color="green", line_width=2)
p1.line(ind, Softmax_absulate, legend="Softmax Absolute", line_color="green", line_width=2)
#p1.arc(ind, AnnealingSoftmax, legend="Annealing Softmax", line_color="grey", end_angle=90)
p1.circle(ind, AnnealingSoftmax_absulate, legend="Annealing Softmax Absolute", line_color="grey")
p1.line(ind, AnnealingSoftmax_absulate, legend="Annealing Softmax Absolute", line_color="grey")
#p1.oval(ind, Exp3, legend="Exp3", line_color="black", height=0.01, width=0.01)
p1.circle(ind, Exp3_absulate, legend="Exp3 Absolute", line_color="black", line_width=2)
p1.line(ind, Exp3_absulate, legend="Exp3 Absolute", line_color="black", line_width=2)
#p1.arc(ind, Hedge, legend="Hedge", line_color="yellow")
#p1.triangle(ind, Hedge, legend="Hedge", line_color="yellow")
p1.circle(ind, Hedge_absulate, legend="Hedge Absolute", line_color="yellow", line_width=2)
p1.line(ind, Hedge_absulate, legend="Hedge Absolute", line_color="yellow", line_width=2)
#p1.square_cross(ind, ThompsonSampling, legend="Thompson Sampling", line_color="pink")
p1.circle(ind, ThompsonSampling_absulate, legend="Thompson Sampling Absolute", line_color="pink")
p1.line(ind, ThompsonSampling_absulate, legend="Thompson Sampling Absolute", line_color="pink")
p1.line(ind, Acc, legend="Accuracy", line_dash=(4, 4), line_color="orange", line_width=2)
#p1.square(ind, Hedge, legend="3*sin(x)", fill_color=None, line_color="brown")
p1.title.align = "center"
show(p1)
#show(gridplot(p1, p2, ncols=2, plot_width=400, plot_height=400)) # open a browser

```

1.4 UCB1

```

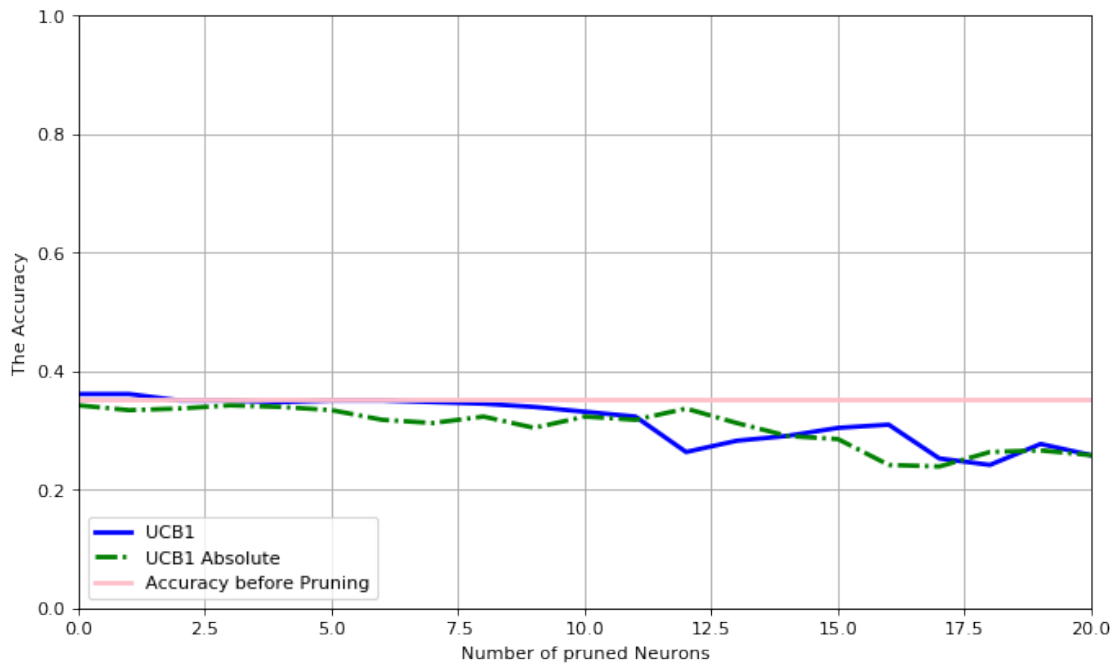
In [25]: fig = plt.figure(figsize=(10, 6), dpi=80)
ax = fig.add_subplot(111)
N = len(ucb1)
Acc = [Accuracy for col in range(N)]

```

```

## necessary variables
ind = np.arange(N) # the x locations for the groups
plt.plot(ind , ucb1 , color="blue", linewidth=2.5, linestyle="-", label="UCB1")
plt.plot(ind , ucb1_absulate , color="green", linewidth=2.5, linestyle="-.", label="UCB1 Absolute")
plt.plot(ind , Acc, color="pink", linewidth=2.5, linestyle="-", label="Accuracy before pruning")
plt.legend(loc = 3)
plt.axis([0, 20, 0, 1])
plt.xlabel('Number of pruned Neurons')
plt.ylabel('The Accuracy')
plt.grid(True)
plt.show()

```



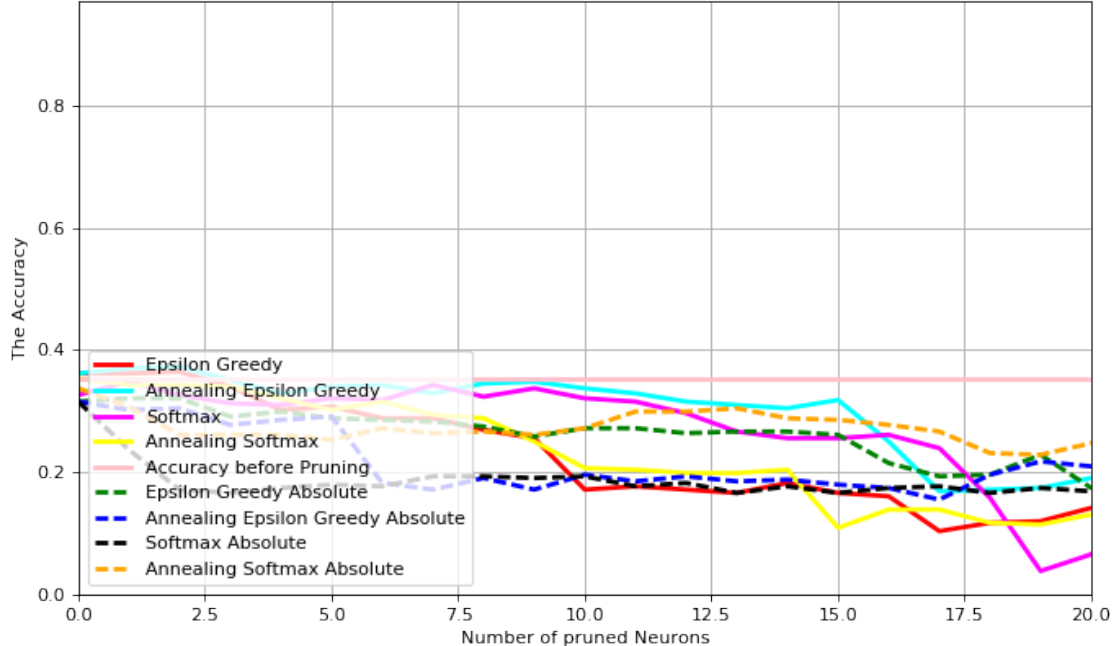
```

In [26]: p1 = figure(title="The Performance over the number of neurons' pruned", tools=TOOLS)
        #p1.circle(ind, ucb1, legend="ucb1", color="orange")
        p1.line(ind, ucb1, legend="ucb1", line_color="blue", line_width=2)
        p1.circle(ind, ucb1_absulate, legend="ucb1 Absolute", line_color="green", line_width=2)
        p1.line(ind, ucb1_absulate, legend="ucb1 Absolute", line_color="green", line_width=2)
        p1.line(ind, Acc, legend="Accuracy", line_dash=(4, 4), line_color="orange", line_width=2)
        #p1.square(ind, Hedge, legend="3*sin(x)", fill_color=None, line_color="brown")
        p1.title.align = "center"
        show(p1)
        #show(gridplot(p1, p2, ncols=2, plot_width=400, plot_height=400)) # open a browser

```


1.5 Epsilon greedy and Softmax

```
In [27]: fig = plt.figure(figsize=(10, 6), dpi=80)
ax = fig.add_subplot(111)
N = len(EpsilonGreedy)
Acc = [Accuracy for col in range(N)]
## necessary variables
ind = np.arange(N) # the x locations for the groups
plt.plot(ind, EpsilonGreedy, color="red", linewidth=2.5, linestyle="-", label="EpsilonGreedy")
plt.plot(ind, AnnealingEpsilonGreedy, color="cyan", linewidth=2.5, linestyle="-", label="AnnealingEpsilonGreedy")
plt.plot(ind, Softmax, color="magenta", linewidth=2.5, linestyle="-", label="Softmax")
plt.plot(ind, AnnealingSoftmax, color="yellow", linewidth=2.5, linestyle="-", label="AnnealingSoftmax")
plt.plot(ind, Acc, color="pink", linewidth=2.5, linestyle="-", label="Accuracy before Pruning")
plt.plot(ind, EpsilonGreedy_absolute, color="green", linewidth=2.5, linestyle="--", label="EpsilonGreedy Absolute")
plt.plot(ind, AnnealingEpsilonGreedy_absolute, color="blue", linewidth=2.5, linestyle="--", label="AnnealingEpsilonGreedy Absolute")
plt.plot(ind, Softmax_absolute, color="black", linewidth=2.5, linestyle="--", label="Softmax Absolute")
plt.plot(ind, AnnealingSoftmax_absolute, color="orange", linewidth=2.5, linestyle="--", label="AnnealingSoftmax Absolute")
plt.legend(loc = 3)
plt.axis([0, 20, 0, 0.97])
plt.xlabel('Number of pruned Neurons')
plt.ylabel('The Accuracy')
plt.grid(True)
plt.show()
```



```
In [28]: p1 = figure(title="The Performance over the number of neurons' pruned", tools=TOOLS)
#p1.square(ind, EpsilonGreedy, legend="Epsilon Greedy", fill_color=None, line_color="red")
```



```

p1.line(ind, EpsilonGreedy, legend="Epsilon Greedy", line_color="red", line_width=2)
#p1.ellipse(ind, AnnealingEpsilonGreedy, legend="Annealing Epsilon Greedy", line_color="blue")
p1.line(ind, AnnealingEpsilonGreedy, legend="Annealing Epsilon Greedy", line_color="blue")
#p1.diamond(ind, Softmax, legend="Softmax", line_color="green")
p1.line(ind, Softmax, legend="Softmax", line_color="green", line_width=2)
#p1.arc(ind, AnnealingSoftmax, legend="Annealing Softmax", line_color="grey", end_angle=90)
p1.line(ind, AnnealingSoftmax, legend="Annealing Softmax", line_color="grey", line_width=2)
p1.circle(ind, EpsilonGreedy_absulate, legend="Epsilon Greedy Absolute", line_color="orange")
p1.line(ind, EpsilonGreedy_absulate, legend="Epsilon Greedy Absolute", line_color="orange")
#p1.ellipse(ind, AnnealingEpsilonGreedy, legend="Annealing Epsilon Greedy", line_color="blue")
p1.circle(ind, AnnealingEpsilonGreedy_absulate, legend="Annealing Epsilon Greedy Absolute", line_color="blue")
p1.line(ind, AnnealingEpsilonGreedy_absulate, legend="Annealing Epsilon Greedy Absolute", line_color="blue")
#p1.diamond(ind, Softmax, legend="Softmax", line_color="green")
p1.circle(ind, Softmax_absulate, legend="Softmax Absolute", line_color="cyan", line_width=2)
p1.line(ind, Softmax_absulate, legend="Softmax Absolute", line_color="cyan", line_width=2)
#p1.arc(ind, AnnealingSoftmax, legend="Annealing Softmax", line_color="grey", end_angle=90)
p1.circle(ind, AnnealingSoftmax_absulate, legend="Annealing Softmax Absolute", line_color="grey")
p1.line(ind, AnnealingSoftmax_absulate, legend="Annealing Softmax Absolute", line_color="grey")
p1.line(ind, Acc, legend="Accuracy", line_dash=(4, 4), line_color="black", line_width=2)
#p1.square(ind, Hedge, legend="3*sin(x)", fill_color=None, line_color="brown")
p1.title.align = "center"
show(p1)
#show(gridplot(p1, p2, ncols=2, plot_width=400, plot_height=400)) # open a browser

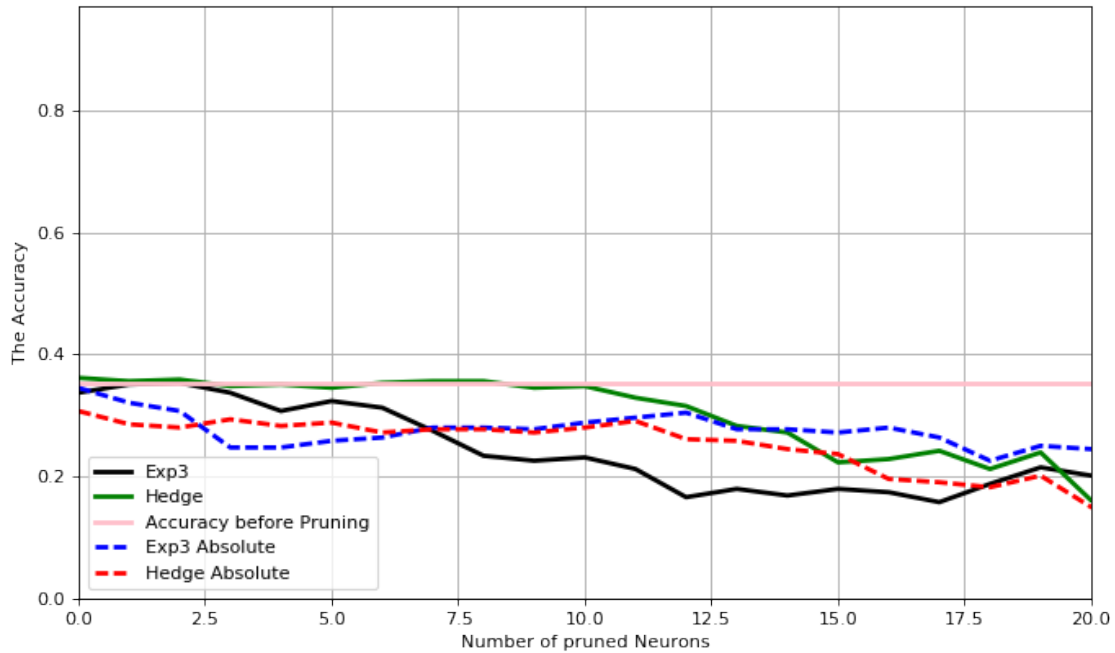
```

1.6 Adversial Bandits Hedge and EXP3

```

In [29]: fig = plt.figure(figsize=(10, 6), dpi=80)
ax = fig.add_subplot(111)
N = len(Exp3)
Acc = [Accuracy for col in range(N)]
## necessary variables
ind = np.arange(N) # the x locations for the groups
plt.plot(ind, Exp3, color="black", linewidth=2.5, linestyle="-", label="Exp3")
plt.plot(ind, Hedge, color="green", linewidth=2.5, linestyle="-", label="Hedge")
plt.plot(ind, Acc, color="pink", linewidth=2.5, linestyle="-", label="Accuracy before")
plt.plot(ind, Exp3_absulate, color="blue", linewidth=2.5, linestyle="--", label="Exp3")
plt.plot(ind, Hedge_absulate, color="red", linewidth=2.5, linestyle="--", label="Hedge")
plt.legend(loc = 3)
plt.axis([0, 20, 0, 0.97])
plt.xlabel('Number of pruned Neurons')
plt.ylabel('The Accuracy')
plt.grid(True)
plt.show()

```



```
In [30]: p1 = figure(title="The Performance over the number of neurons' pruned", tools=TOOLS)
#p1.oval(ind, Exp3, legend="Exp3", line_color="black", height=0.01, width=0.01)
p1.line(ind, Exp3, legend="Exp3", line_color="black", line_width=2)
#p1.arc(ind, Hedge, legend="Hedge", line_color="yellow")
#p1.triangle(ind, Hedge, legend="Hedge", line_color="yellow")
p1.line(ind, Hedge, legend="Hedge", line_color="yellow", line_width=2)
p1.circle(ind, Exp3_absolute, legend="Exp3 Absolute", line_color="green", line_width=2)
p1.line(ind, Exp3_absolute, legend="Exp3 Absolute", line_color="green", line_width=2)
#p1.arc(ind, Hedge, legend="Hedge", line_color="yellow")
#p1.triangle(ind, Hedge, legend="Hedge", line_color="yellow")
p1.circle(ind, Hedge_absolute, legend="Hedge Absolute", line_color="red", line_width=2)
p1.line(ind, Hedge_absolute, legend="Hedge Absolute", line_color="red", line_width=2)
p1.line(ind, Acc, legend="Accuracy", line_dash=(4, 4), line_color="orange", line_width=2)
#p1.square(ind, Hedge, legend="3*sin(x)", fill_color=None, line_color="brown")
p1.title.align = "center"
show(p1)
#show(gridplot(p1, p2, ncols=2, plot_width=400, plot_height=400)) # open a browser
```

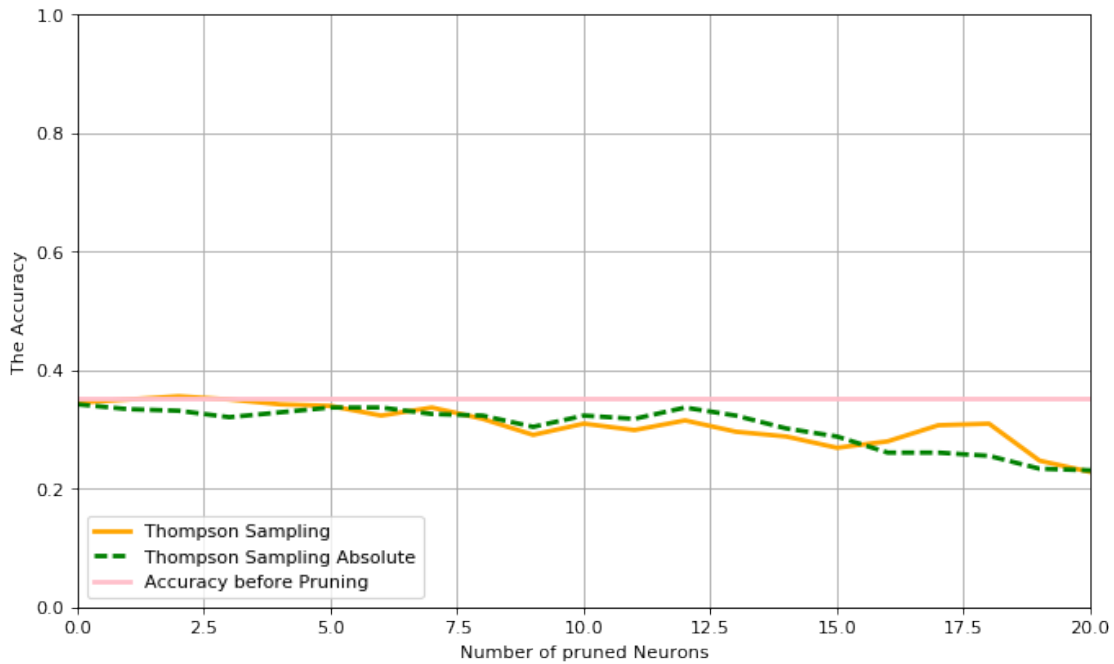
2 Thompson Sampling

```
In [31]: fig = plt.figure(figsize=(10, 6), dpi=80)
ax = fig.add_subplot(111)
N = len(ThompsonSampling)
Acc = [Accuracy for col in range(N)]
## necessary variables
```

```

ind = np.arange(N)                                # the x locations for the groups
plt.plot(ind , ThompsonSampling, color="orange", linewidth=2.5, linestyle="-", label="T")
plt.plot(ind , ThompsonSampling_absulate, color="green", linewidth=2.5, linestyle="--", label="T")
plt.plot(ind , Acc, color="pink", linewidth=2.5, linestyle="-", label="Accuracy before")
plt.legend(loc = 3)
plt.axis([0, 20, 0, 1])
plt.xlabel('Number of pruned Neurons')
plt.ylabel('The Accuracy')
plt.grid(True)
plt.show()

```



```

In [32]: p1 = figure(title="The Performance over the number of neurons' pruned", tools=TOOLS)
        #p1.square_cross(ind, ThompsonSampling, legend="Thompson Sampling", line_color="pink")
        p1.line(ind, ThompsonSampling, legend="Thompson Sampling", line_color="pink", line_widt
        p1.line(ind, Acc, legend="Accuracy", line_dash=(4, 4), line_color="orange", line_width=
        p1.circle(ind, ThompsonSampling_absulate, legend="Thompson Sampling Absolute", line_col
        p1.line(ind, ThompsonSampling_absulate, legend="Thompson Sampling Absolute", line_color
        #p1.square(ind, Hedge, legend="3*sin(x)", fill_color=None, line_color="brown")
        p1.title.align = "center"
        show(p1)
        #show(gridplot(p1, p2, ncols=2, plot_width=400, plot_height=400)) # open a browser

```