

# Постепенно увеличиваем разрешение с ProgressiveGridSearch

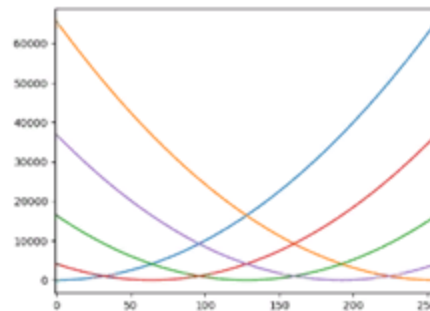
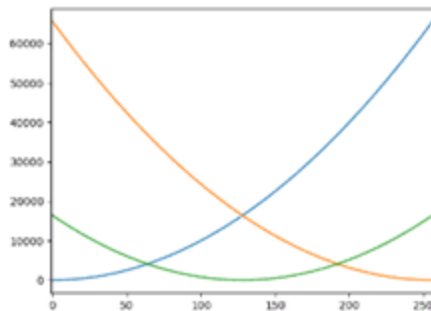


# Оптимизируем порядок вычислений

Идея строится на том, что если достаточно разумно выбрать порядок узлов, то большая часть артефактов ландшафта функции будет заметна уже на 30-70% вычислений

$$x^{k+1} = \arg \max_{x \in H} \{distance(x^l, x)\}_{l=1}^k$$

2

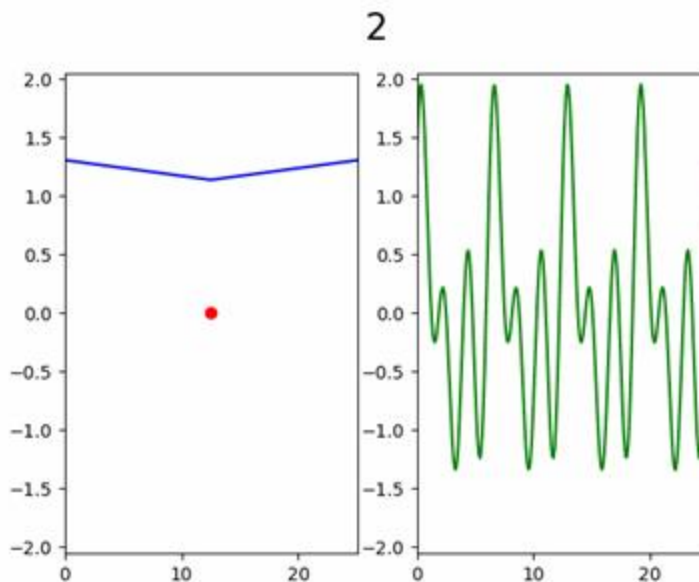


Графики  $(x-x^l)^2$  для  $k=3$  и  $k=5$



# Проверяем гипотезу

Задали частичную сумму  
ряда Фурье со случайными  
коэффициентами



# Что делать если переменных больше одной?

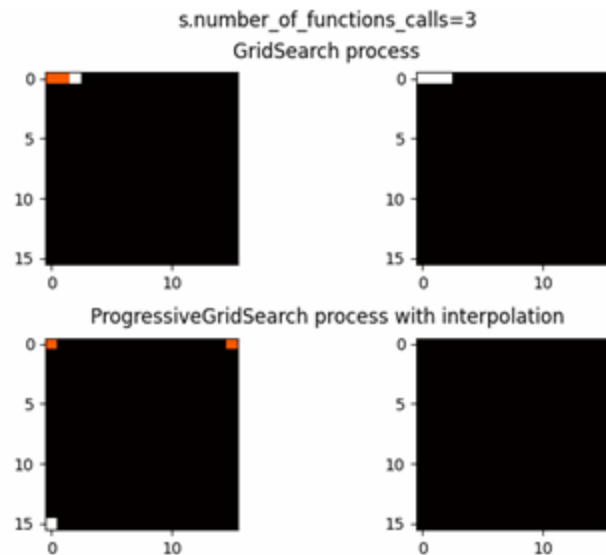
$$x^{k+1} = \arg \max_{x \in H} \{distance(x^l, x)\}_{l=1}^k$$

Оказывается решением этой многокритериальной оптимизационной задачи являются вершины гиперкубов вложенных в гиперкуб, который является множеством возможных значений (то есть каждый квадрат делится на 4 квадрата, а каждый из этих 4х квадратов на еще 4 квадрата, а каждый из этих 16ти на еще 4 квадрата и так далее)



# Сравнение

Здесь заданы две целые переменные (row и column). И, как видно, пока GridSearch пробегает одну строчку ProgressiveGridSearch уже довольно детально получил представление о ландшафте функции



# Сравнение на более сложном ландшафте

optimizer.number\_of\_functions\_calls=3

GridSearch process



ProgressiveGridSearch process with interpolation

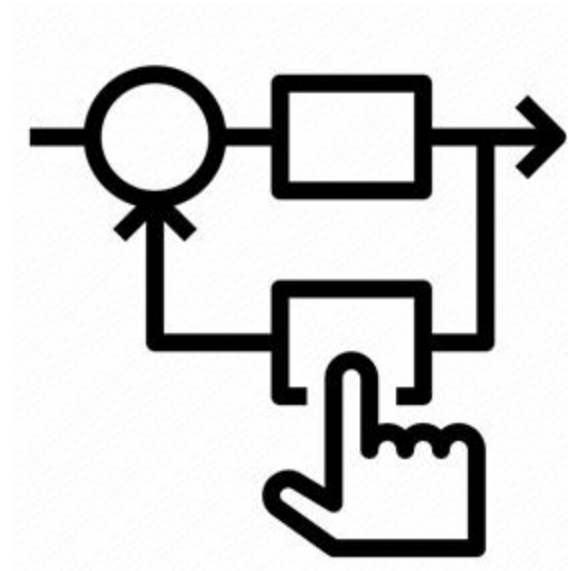


# Заключение выступления и начало дискуссии

Зовут меня Николай Стрекопытов, можно на ты

Канал о моих разработках [t.me/research\\_and\\_deep\\_learning](https://t.me/research_and_deep_learning)

Там можно взять и мой контакт в тг



[vk.cc/cmTvKE](https://vk.cc/cmTvKE)

