Proposal

## Notation

* M: number of filters
* L: number of layers
* P: number of parameters
* E: ensemble size

## Initial assumptions

* Since authors observed that the performance is insensitive to M, we won’t take it into account
* We define training budget in terms of:
  + Hardware: Capacity of a network (# parameters) / Memory Consumption (≈ feature maps)
  + Training / Inference time
* The limit on the number of layers for a constant P is only constraint by the inference time (\*)
* All the hidden layers have the same size

## Previous statements

* Allocating a fixed number of parameters across multiple layers is better than putting them in few layers
* Filter parameters are best allocated in multilayer stacks

## Our approach

The question we want to answer are:

1. If you have a fixed budget (scenario: you have a small hardware device), what is the best to do?
   1. Comparative between plain models and recursive models
   2. Answer: for this particular budget, the best model is option X
      1. The dimensions of a scenario are training time / inference time / test accuracy
2. Does recursivity behave the same way in all the approaches? We are interested in knowing if recursivity (as a possibility to increase performance without exceeding the budget) works equal in the different regimes
   1. First, use recursive methods in every alternative as a simulation to what would happen
   2. Then, contrast that hypothesis by really increasing the budget in all the alternatives

Are there different regimes of improvement behaviour?

1. Are there phenomena dependent on the network architecture? (Is it only for ANN, CNN…)
2. Are these phenomena dependent on the difficulty on the task? To ensure experiments are not altered by overfitting, ...

We have a strong intuition that ensemble models will perform better (based on results obtained for ResNets). This will set a good baseline to encourage ensemble learning algorithm like Mother Nets.

If True as a complementation, if False, as a revalidation from ensembles

1. Apply techniques of ensemble learning to improve their scenario (training time, inference time, test accuracy) dimensions and outperform previous results.

## Models

