# Exercise 4 Analysis of a Deep Architecture

Residual connections are crucial for training very deep architectures. Since Inception networks are typically very deep, it is natural to replace the filter chaining stage of the Inception architecture with residual connections. In this way, Inception could take full advantage of the residual approach while maintaining its computational efficiency. However, the use of residual links appears to significantly improve training speed, which in itself is a good argument for their use.

It is reasonable to expect that consistent use of batch normalization should be beneficial, but Szegedy et al. wanted each model replica to be trained on a single GPU. It turned out that the memory requirements of layers with large activation sizes took up a disproportionate amount of GPU memory. By omitting batch normalization on these layers, Szegedy et al. were able to significantly increase the total number of inception blocks.

Szegedy et al. also found that when the number of filters exceeded 1000, the residuals exhibited instabilities and the network "died" early in training, meaning that the last layer before average pooling produced only zeros after a few tens of thousands of iterations. By reducing the size of the residuals before adding them to activate the previous layer, they were able to stabilize the training. In addition they investigated how introducing residual connections dramatically improved the training speed for the inception architecture.

## Comparison

I don’t really think we can compare the inception network with the those “simple” deeper networks from the lecture and exercise, because we’ve only strain. The Inception architecture its somehow more complex and the managed to split via a filter concat the main strain into two and afterwards to merge it. I think it would be really interesting to see how to build such a architecture and what are the benefits of it!

# Exercise 5 Optional : Review Questions

1. Explain the notion of hierarchical features with CNNs.
   1. the deeper the layers become the more important the details become
2. Explain 2 strategies to visualise the modelling taking place in CNNs.
   1. Extract output of each layer from the CNN
   2. Rebuild new Model in revised order
3. What do we try to fight when using data augmentation ?
   1. By applying data augmentation we can increase the ability of our model to generalize and make better, more accurate predictions on data it was not trained on.
4. What are the implementation strategies for data augmentation ?
   1. It has to be realistic, and we have to take care not do augment something what does not exist.
5. Explain the main differences for the deep architectures seen in class : AlexNet, VGGNet, GoogLeNet, ResNet. What were their intuitions when putting together such architectures?
   1. Different number of parameters, and accordantly diff time to train and reducing accuracy
   2. The more deeper the more accurate will be the prediction