Problem 1

- Conduct the experiments suggested in “3.5-classifying-newswires” and report the results with discussions.

Further experiments

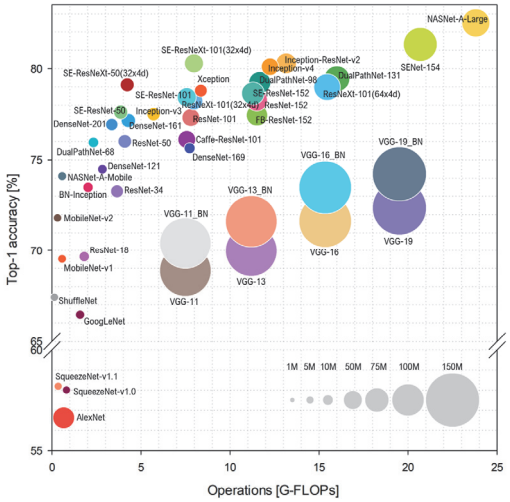
- Try using larger or smaller layers: 32 units, 128 units...
- We were using two hidden layers. Now try to use a single hidden layer, or three hidden layers.

Problem 2 (Foreword)

- You know that there are many (I would say, too many) methods of machine learning, today. In the field of deep learning, it holds, too.
- Keras package provide us a chance to use some of them very easily.
  - https://keras.io/applications/
  - For R interface, refer to https://www.rdocumentation.org/packages/keras/versions/2.2.5.0 (the set of the DNN models is the same)
- The figure in the next slide is from a paper:

  Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. IEEE Access 6 (2018), 64270--64277.

- Please choose at least three of the models other than VGG16 to complete your report.



https://ieeexplore.ieee.org/document/8506339

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.79	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.9	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.76	0.93	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.78	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.75	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.96	88,949,818	-

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset

https://keras.io/applications/

Problem 2 (cont.)

- Try “5.3-using-a-pretrained-convnet” with base models you like and report results with discussions.
  - Note that you have to edit “4 * 4 * 512” and “4, 4, 512” appropriately as suggested in the corresponding comments.
  - Also you have to edit the layer name in

    unfreeze_weights(conv_base, from = "block3_conv1")

  - Not also that you may choose GPU or TPU for the accelerator.