Author: Derek Walton

VGG vs ResNet (Overview)

Background → ImageNet Challenge

ImageNet is a large visual database designed for use in visual object recognition research. More than 14 million images have been hand annotated by the project to indicate what objects are pictured. It contains more than 20,000 categories, with each category (i.e., strawberry, balloon, etc.) consisting of at least several hundred images. Since 2010 it has been used for an annual contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where software programs compete to correctly classify and detect objects and scenes; the contest uses a "trimmed" list of 1,000 non-overlapping classes.

Competetion Parameters:

Top 5 Error rate:

- The model has 5 chances to correctly identify the label

Top 1 Error rate:

- The model has a single chance to correctly identify the label
- ~20% decrease in accuracy in all models compared to the Top 5

Initial goal of the competition was to beat the human error rate

Human Error rate for ImageNet:

- 5.1%

VGGNet (2014): 11.2% to 7.3% Error rate

- Produced by Oxford University
- Beautifully uniform: 3x3 conv, stride 1, pad 1, 2x2 max pool
- 16 layers
- 138 million parameters

ResNet (2015): 6.7% to 3.57% Error Rate

- Produced by Microsoft
- More layers = better performance
    - Not always true, uses skip connections to greatly reduce the effect of Vanishing/Exploding gradient (gradient becomes 0 or too large)

- Skip connections provide block deeper in the network with the unmodified input data in addition to the output from the previous layer
- Any layers that produce output that would be harmful to the model are skipped by comparing the two potential inputs

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9	3.8
DenseNet121	33	75.0%	92.3%	8.1M	242	77.1	5.4
DenseNet169	57	76.2%	93.2%	14.3M	338	96.4	6.3
DenseNet201	80	77.3%	93.6%	20.2M	402	127.2	6.7
NASNetMobile	23	74.4%	91.9%	5.3M	389	27.0	6.7
NASNetLarge	343	82.5%	96.0%	88.9M	533	344.5	20.0
EfficientNetB0	29	77.1%	93.3%	5.3M	132	46.0	4.9
EfficientNetB1	31	79.1%	94.4%	7.9M	186	60.2	5.6
EfficientNetB2	36	80.1%	94.9%	9.2M	186	80.8	6.5
EfficientNetB3	48	81.6%	95.7%	12.3M	210	140.0	8.8
EfficientNetB4	75	82.9%	96.4%	19.5M	258	308.3	15.1
EfficientNetB5	118	83.6%	96.7%	30.6M	312	579.2	25.3

- *Image taken from keras documentation page:* https://keras.io/api/applications/

Bottom Line: ResNet has shown slightly better accuracy identifying objects, however it was trained using ~1/7 the number of parameters as VGG. This is not necessarily a bad thing:

Parameters are crucial for deep learning models as they capture the knowledge and patterns in the training data. They represent the numerical values that determine the behavior of the model and enable it to make predictions on new, unseen data.

In the case of ResNet and VGG, the parameters typically include convolutional filters, fully connected layer weights, and bias terms. These parameters are optimized using gradient-based optimization algorithms such as stochastic gradient descent (SGD) or its variants.

The number of parameters in a model is an important factor as it affects the model's complexity, memory requirements, and training time. Larger models with more parameters can potentially capture more complex patterns but may also be more prone to overfitting if the training data is limited.

Sources:

1). https://en.wikipedia.org/wiki/ImageNet

2). https://www.youtube.com/watch?v=HdlQOJe_mU0

3). https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/