Report

Experiment 4:

DenseNet169: Key Ideas over DenseNet169 modelling approach

Densenet models with weights pre-trained on ImageNet. 169 denotes the depth of the ImageNet models. DenseNet169 is a network architecture where each layer is directly connected to every other layer in a feed-forward fashion (within each dense block). The default input size for this model is 224x224. A 5-layer dense block with a growth rate of k = 4

Recent work has shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. In this paper, we embrace this observation and introduce the Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

Counter-intuitive effect of this dense connectivity pattern is that it requires fewer parameters than traditional convolutional networks, as there is no need to relearn redundant feature maps. Traditional feed-forward architectures can be viewed as algorithms with a state, which is passed on from layer to layer. Each layer reads the state from its preceding layer and writes to the subsequent layer. It changes the state but also passes on information that needs to be preserved. Our proposed DenseNet architecture explicitly differentiates between information that is added to the network and information that is preserved. DenseNet layers are very narrow (e.g., 12 feature-maps per layer), adding only a small set of feature-maps to the “collective knowledge” of the network and keep the remaining feature-maps unchanged — and the final classifier decides based on all feature-maps in the network.

Besides better parameter efficiency, one big advantage of DenseNets is their improved flow of information and gradients throughout the network, which makes them easy to train. Each layer has direct access to the gradients from the loss function and the original input signal, leading to an implicit deep supervision. This helps training of deeper network architectures. Further, we also observe that dense connections have a regularizing effect, which reduces over- fitting on tasks with smaller training set sizes.

Experiment 2:

Autokeras provides functions to automatically search for architecture and hyperparameters of deep learning models. We did three experiments to reproduce it’s functionality over 32 x 32 tiny imagenet dataset. The first model created here was fixed to a time limit of 12 hours instead of default(24hrs). Each epoch took atleast 30mins of running time and by default it ran upto more than 150 epochs.

After Training model 0 it produced a result of no loss decrease after 5 epochs. The loss value was 13.3 and metric Value was of around 0.232. The next traning model 1 ran for couple of epochs and automatically stopped on RAM crash.

For the next experiment we reduced the time limit to an hour with epochs reduced to 7. At this point Training model 0 created a loss value of 15.11 and Metric value of 0.15. The next model training got interrupted again and gave a result of no loss decrease after 30 epochs. A prediction value of 27.16% was the result.