AlphaNet: Architecture, Models, and Applications

Final Presentation, June 2020
Major Project (CSE–419 and CSE–429)
MANIT, Bhopal (India) – 462003

Proposed Work

- Theoritical works
  - Alpha-net: Framework, Design Architecture
    - Improvements over traditional RESNETs (RESNET-32, etc.)
    - For plain networks and block networks
  - Analysis and Modeling with novelties
- Practical works
  - Implementation of Alpha-Net v1, v2, v3, and v4
  - Quantitative analysis over previous architectures and different techniques

Alpha-Net: Architecture

Alpha transforms (over convoluted layers)

- For normalization of data for lesser feature vector size
  - Stochastic Interconnection between layers
  - Data flow according to Alpha blocks
- Architecture
  - Novel loss function (AM Softmax + Linear weights)
  - Alpha-encoding in preprocessing
  - Diagram (...continued)

Novelties in the project

- Normalization function – Alpha-encoding
  - Better than log-scaling and z-score normalization in accuracy
- Loss function – Additive Margin Softmax + Linear weights
  - Better than Softmax and AM Softmax in accuracy
- Layer interconnection – Stochastic Manner
  - Better than linear interconnection in data per layer complexity
- Dataset – Hybrid benchmark similar to that of ImageNet
  - Better training complexity because of simple and small representation (Alpha-blocks and convolutions)

Alpha-Net: Novel loss function

$$\Lambda_{AMS} = -\frac{1}{n}\sum_{i=1}^{n}\frac{e^{s(cos\theta_{y_i}-m)}}{e^{s(\cos\theta_{y_i}-m)}+\sum_{j=1,j\neq y}^{c}e^{s\cos\theta_j}}$$

For linear weights ($y = ax_1 + c$), we can add a piece-wise division of it to make AM Softmax function with linear weights trained on ImagNet benchmark, as follows:

$$\Lambda_{AMS} = \begin{cases} -\frac{1}{n}\sum_{i=1}^{n}\frac{e^{s(cos\theta_{y_i}-m)}}{e^{s(\cos\theta_{y_i}-m)}+\sum_{j=1,j\neq y}^{c}e^{s\cos\theta_j}} & \theta - m > 0 \\ ax_i + c & \theta - m <= 0 \end{cases}$$

Where:

Λ_{AMS} = Calculated Loss
n = Training instances
θ = Angle with the origin
m = Gradient for the instance
s = Sample value of the current instance
a and c = Linear weight coefficients

Excerpt from report

Conclusion

- Designed Alpha-Net: Architecture with novel loss function, layer arrangements, normalization function, etc.
- Implemented 4 models – Alpha-Net v1, v2, v3, and v4 based on Alpha-Net architecture with 128, 256, 512, and 1024 layers.
- Novel loss function: Additive margin Softmax + Linear weights.
- Quantitative analysis of Top 1 Accuracy with different layer structures, different normalization functions, and differnt loss functions.
- Comparisons with different architectures previously proposed.
- Custom dataset with ImageNet benchmark as an inspiration + large size
- Application of network ranges to a broad set of training tasks.