Hyperactivations for Activation Function Exploration

Gobind Puniani
MATH 198: Spring 2020
02/06/20

Activation Functions

- Usually a fixed activation function used to introduce non-linearity between layers
- Often ignored as researchers stick with popular choices
  - ReLU, Tanh, etc.
- Can affect network's capacity, speed of convergence, and performance accuracy
  - Varied activations may have greater effect

Activation Functions (Cont.)

- Currently unknown as to how activation functions interact to produce resulting behavior
- Performance of activations depends on architecture and task (despite claims to the contrary for new ones)
- Idea: let the network find the best activation function during training

Previous Methods

- Agostellini et al. proposed developing activations for each individual neuron
  - Rough function surfaces(?)
  - Overnormalized
- Use reinforcement learning to generate non-linearities from training many networks
  - Reward based on performance of activation
  - Swish invented this way
  - Limited search space due to restricted math operators
  - Extreme computational cost
  - Generalized, not task-specific

Hyperactivations

- Authors opted for learned activations per layer due for computational and parameter efficiency
- One hyperactivation takes the place of all activations
- Hyperactivation constructed from two parts:
  - Activation network (shallow forward-feed net)
  - Hypernetwork
- Activation network needs a function to bootstrap from

Important Equations

- A is the nonlinearity, W_a is the activation weight matrix, x is the vector, and e is the embedding
- With the vector and reshaping:
  $AN(x) = \text{reshape}(A(vec(x), W_a), x_{(w_j, h_j)})$
- Incorporating a hypernetwork in activation network above:
  $AN(x) = \text{reshape}(A(vec(x), H(e, W_h)), x_{(w_j, h_j)})$

Experiment

- MNIST
  - Small CNN trained with Adam
  - Smaller network: non-linearities more non-linear
  - Learned activations far different from popular choices
- CIFAR-10
  - ResNet-16 with 9 ReLUs replaced by hyperactivations
  - Trained with Adam for only 10 epochs

Conclusion

- First to consider a hypernetwork-based meta-learning approach to learning activation functions
- Hyperactivations produced activation functions better suited for the tasks than standard activation functions
    - Faster convergence and higher test accuracy in both MNIST and CIFAR-10