Layer gain $G_i = \text{var}(L_i) / \text{var}(L_{i-1})$, where $\text{var}(L_i)$ – variance of layer input, $\text{var}(L_{i-1})$ – variance of layer output.

1. Very deep neural networks are powerful but hard to train.
2. Observation: regardless of the non-linearity used, deep net trains well if its product of per-layer gains equals to one: $G_{\text{NN}} = G_1 \times \cdots \times G_i = 1$ (1)
3. Initiative satisfying Eq. (1) exists only for linear and ReLU networks. We propose an initialization algorithm applicable to any feedforward network.

Pre-initialize network with orthonormal matrices as in Saxe et.al. (2013) for each convolutional and fully-connected layer $L$ do

- do forward pass with mini-batch
- calculate $\text{var}(O_L)$
- $\frac{\text{W}^{2}_{\text{L}}}{\text{W}^{2}_{\text{L}}} = \sqrt{\text{var}(O_L)}$
- until $\text{var}(O_L) \in (0, 1)$ or $(T, T_{\text{max}})$ end for

The LSV algorithm does not deal with biases and initializes them with zero.

Very deep neural networks are powerful but hard to train. (1)

Common initialization methods lead to the same layer gain $G_i$ and fully-connected layer, which works if input variance is 1 and no other types of layers are present. If a significant number of other type of layers is present:

- Layer gain $G_i < 1$ – vanishing variance
- Layer gain $G_i > 1$ – exploding variance

Deriving layer gain $G_i$ proportionally ensuring $\prod_{i=1}^{L} G_i = 1$ is a hard task for general network with various activation functions, poolings, skip connections, etc.

Machine learning basics: centered and normalized (mean = 0, var = 1) is input good. Glorot & Bengio (2010): keep input [and output] of each layer normalized, propose weight initialization formula for linear net. modifies the Glorot formula for ReLU net.


- Recurring theme: all you need is a good init

The authors were supported by The Czech Science Foundation Project GACR P103/12/G084 and CTU student grant SGS15/155/OHK3/2T.

REFERENCES