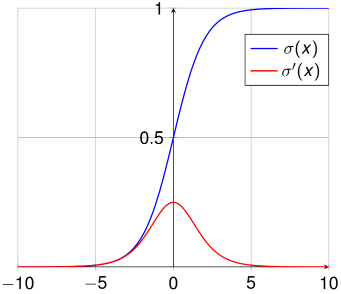
Machine Learning in Geophysics

Lecture 9 – Neural networks, gradients, activation functions

Gradients



- We have seen how the learning rate μ can lead to instabilities (exploding gradients) or stop convergence (vanishing gradients).
- Deep neural networks can show similar behaviour for other reasons.

We can look at the variance of the outputs z_i in layer i of the network

$$\mathrm{Var}[z_i] = \mathrm{Var}[x] \prod_{j=0}^{i-1} n_j \mathrm{Var}[W_j]$$

assuming all weights W_j have been initialized with the same variance and we are in the linear regime of the activation function.

Question

What is $\mathrm{Var}[z_i]$ if we initialize our weights with $\mathrm{Var}[W_j] = 1$ and all layers have the same size n?

We can look at the variance of the outputs z_i in layer i of the network

$$\text{Var}[z_i] = \text{Var}[x] \prod_{j=0}^{i-1} n_j \text{Var}[W_j]$$

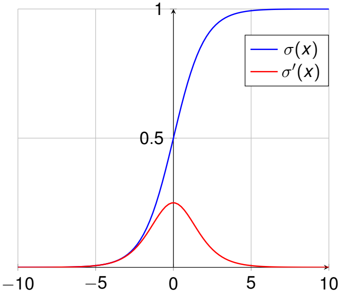
assuming all weights W_j have been initialized with the same variance and we are in the linear regime of the activation function.

Question

What is $\text{Var}[z_i]$ if we initialize our weights with $\text{Var}[W_j] = 1$ and all layers have the same size n?

Answer

$$\text{Var}[z_i] = \text{Var}[x] \prod_{j=0}^{i-1} n = \text{Var}[x] n^{(i-1)}$$

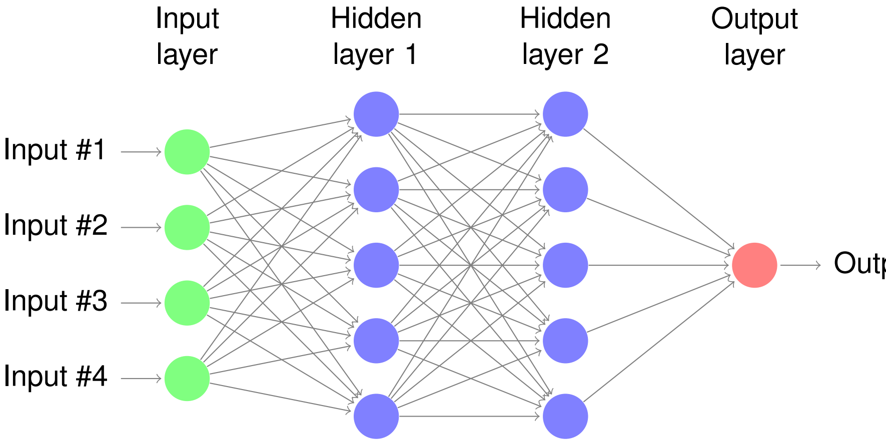


What does

$$\mathrm{Var}[z_i] = \mathrm{Var}[x]n^i$$

mean ?

- The variance of outputs increases in magnitude with each layer.
- Can eventually reach saturation of activation function.
- At saturation gradients become small.

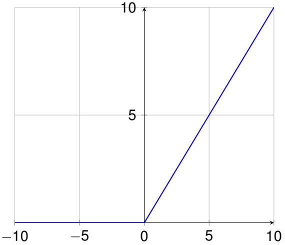

Input layer
Hidden layer 1
Hidden layer 2
Output layer
Input #1
Input #2
Input #3
Input #4
Outp

- How we initialize weights is important.
- Need to maintain variance in deep network
- For sigmoid activation normal distribution with

$$\mathrm{Var}[W_i] = \frac{2}{n_i + n_{i+1}}$$

  is a good strategy.
- Strategy depends on the activation function

ReLU – Rectified Linear Unit

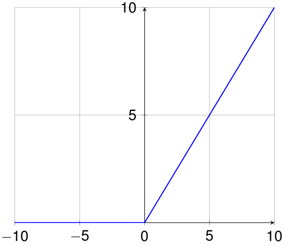


ReLU activation function does not saturate for positive values.

Question

What happens to output and gradient when all inputs are negative?

ReLU – Rectified Linear Unit



ReLU activation function does not saturate for positive values.

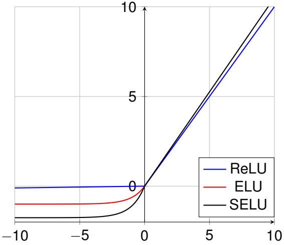
Question

What happens to output and gradient when all inputs are negative?

Answer

Both = 0.

Variants of ReLU



- If weighted sum of inputs is negative for all training data a ReLU neuron is dead.
- Unlikely to be re-activated
- Different variants exist with non-vanishing derivatives

SELU – Scaled exponential linear unit

$$\varphi(z) = \begin{cases} \lambda\left(\alpha \exp(z) - \alpha\right), & z < 0 \\ \lambda z, & z \geq 0 \end{cases}$$

- Special variant of ELU, $\alpha = 1.0507009873554804934\ldots$, $\lambda = 1.6732632423543772\ldots$
- If inputs are normalized (mean = 0, $\sigma = 1$), weights are initialized properly, and network is dense, then NN is self-normalizing
- Solves problem of exploding and vanishing gradients