

Applause from Ludovic Benistant, Mate Labs, and 39 others

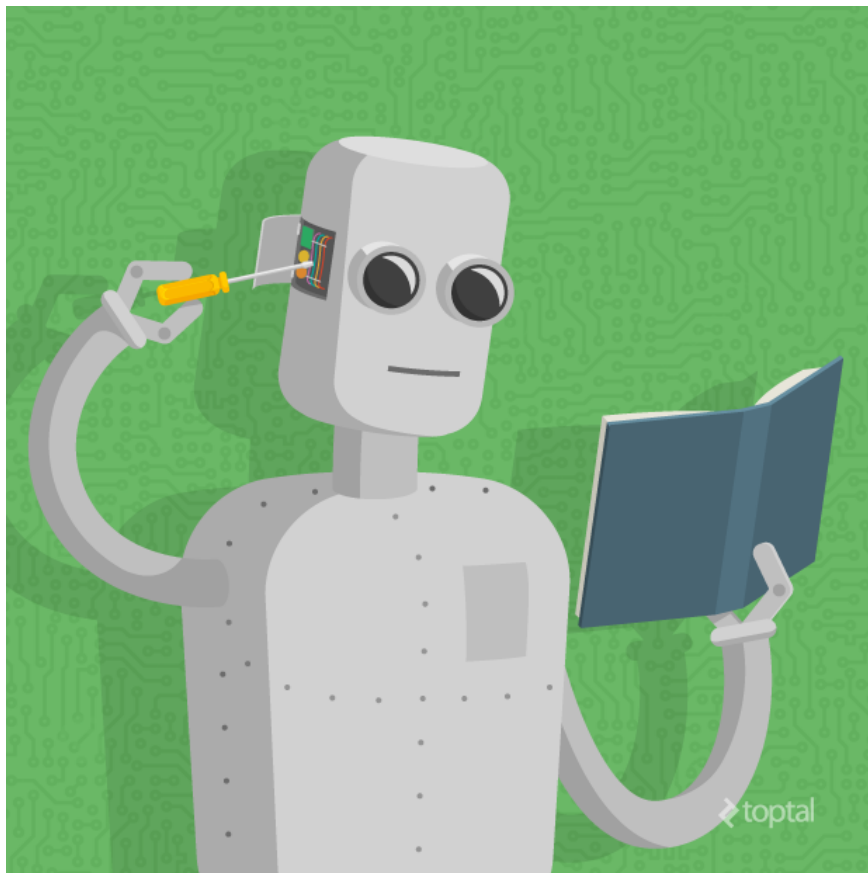


Kailash Ahirwar

Co-founder @ Mate Labs | Democratizing Artificial Intelligence

Jan 30 · 2 min read

Why we need a better learning algorithm than Backpropagation in Deep Learning



Source — toptal -<https://www.toptal.com/>

We all agree on one thing that Back propagation is a revolutionary learning algorithm. For sure, it has helped us in the training of almost all neural network architectures. With the help of GPUs, backpropagation has reduced months of training time to hours/days of training time. It has allowed an efficient training of neural networks.

I think of two reasons because of which it has gotten this widespread adoption, (1) we didn't have anything better than backpropagation, & (2) it worked. Backpropagation is based on the chain rule of differentiation.

Problem lies in the implementation of the Backpropagation algorithm itself. To calculate gradients of the current layer we need gradients of the next layer, so current layer is locked and we can't calculate gradients until and unless we have gradients for the next layer. If we have 1000s of layers in our network, our 1st layer has to wait till eternity to get its weights updated. First few layers in the neural networks are miserable ones and don't get updated properly. Sometimes, in case of the Sigmoid activation function, when we propagate back, gradient vanishes or explodes.

When we take decisions, we take decisions based on our current observation and our previous learning. Current neural networks or deep learning algorithms are not designed the way we take decisions. Our experience defines our decisions. For example, when we walk we use vision, audio and sensory inputs to take decisions. We use learning from one task to learn other tasks.

Limitations of the Backpropagation algorithm:

- It is slow, all previous layers are locked until gradients for the current layer is calculated
- It suffers from vanishing or exploding gradients problem
- It suffers from overfitting & underfitting problem
- It considers predicted value & actual value only to calculate error and to calculate gradients, related to objective function, partially related to the Backpropagation algorithm
- It doesn't consider the spatial, associative and dis-associative relationship between classes while calculating errors, related to objective function, partially related to the Backpropagation algorithm

DeepMind's synthetic gradients shows a workaround, but it is not a solution. In my opinion, we have to think from scratch and design a new learning algorithm which can learn efficiently and can help our network learn in real time.

Disclaimer: This is my personal opinion and it is solely based on my studies and research. I invite you all to share your thoughts on this.

About Me:

I am a Co-Founder of MateLabs, where we have built Mateverse, a ML Platform which enables everyone to easily build and train Machine Learning Models, without writing a single line of code.