Analysis of “Learning representations by backpropagating errors”

A brief statement of the problems addressed in the paper in my own words.

In my own words the problem addressed in this paper is: “There are not any good and reliable methods that can find the weights of nodes within a net”. That is how I would state the problem that the authors are addressing. I think one quote from the paper succinctly hits the nail on the head. “... if we have the right connections from the input units to the output units, we can always find a representation that will perform the mapping from input to output through these hidden units.”

What I agree with/like in the paper and why.

I really liked how they started the reader off with the simple XOR problem of not being able to build a 2-layer perceptron with 2 inputs and 1 output. They then discussed the formation of and use of their delta rule and then applied it to many problems. I also liked how, in a sense they let their error propagation method speak for itself. They didn’t just say “here is a new shiny” and “your welcome” they applied their method to many problems, and in so doing demonstrated at least a breadth of the capabilities of error propagation.

What I disagree with/dislike in the paper and why.

I am not sure where to put this thought that I have, but it seemed as if they were dismissive of problems with local minima, only because they encountered very little of it. Granted, they explained how with their experiments it was rarely if ever an issue, but they spoke as if it wouldn’t really be a problem if you did a few things correctly. Again, I am not being critical, but I would say that you need to be careful about dismissing any potential problems in papers. We just don’t know how our ideas will be used, or what kinds of problems people will be trying to solve 10, or 20, or even 30+ years down the road.

Any inspirations I found in the paper.

You will have to bear with me one this one, but I was impressed by the work they did kind of resembles a crap-shoot, let me explain. The authors mentioned how “it is interesting that not all weights need be variable”, and “Any number of weights in the network can be fixed”. Later they are talking about choosing a learning rate that “is as large as possible without leading to oscillation”. Then when they are discussing the xor problem they talk about the learning rate in a footnote “... the best way to avoid local minima is probably to use very small values of η” They also mention that the generalized delta rule “does not depend on all of the units having identical activation functions”. Then finally rounding off with “Indeed, we frequently discover these more elegant solutions by giving the system more hidden units than it needs and observing that it does not make use of some of those provided.” All of these quotes combined and taken in conjunction with each other make me want to figuratively throw my hands up and just start rolling dice for every decision I make about to design my own neural networks. It seems as if the main theme of neural networks is to just keep bulldozing and churning the dirt and the data until you can discover by happenstance something useful and reliable.

Please note I am not complaining, not even a little bit, it just feels like a significant departure from standard practices, and theory learned up to this point. In effect we are saying “We have all this CPU and GPU power, let’s just churn and bulldoze until we find something meaningful in the stuff we are churning”.

I really liked this paper, mostly because of how eye opening it was to ideas and practices that are still being discussed today. Thanks for the good read Vladimir, I am looking forward to learning more!