What is "Learning" for Neural Networks

Neural networks have spurred a lot of excitement recently. They are quickly proving to be one of the most promising and practical implementations of AI within technological systems. At Fast Forward Labs, we've been researching and building systems that use neural networks for object recognition in images. You may have seen our prototype pictograph. Our goal has been to really understand this technology both technically and in terms of impact and possibility.

As it turns out, these systems are *extremely* difficult to understand due to their complexity. You may have run into one of Google's efforts to make sense of neural networks through their research blog or their explanatory videos.

Nature of this post

This post is an extension of this effort to improve how we understand neural networks, specifically around how they "learn".

Let's start with two points brought home by the team at Google in the above posts:

1. Neural networks are difficult to interpret so we have to develop specific techniques to gain insight on what they are doing.
2. Certain learning tasks us humans do are *very* hard to achieve in artificial neural networks.

In relation to these points, what I've built is a detailed visualization of how a neural network functions at neuron-by-neuron level, and also how it "learns". If you're already familiar with neural networks or want to follow the rest of the post with a visual cue, please see the visualization here

Neural Network Basics

Let's first be sure that we're all on the same page about neural networks basics. Neural networks are composed of layers of computational units (neurons) where each neuron in a layer is connected to each neuron in the next layer. We pass data (e.g., pixel values in an image, words in a text) into our input layer, then many hidden layers transform this data until the output layer makes a prediction or classification on the original data.

How this "transformation" works is that each neuron passes along a value to the next neuron. During this passing, we multiply the value by some **weight**, sum it up with all the other values incoming to the same neuron, adjust it by the neuron's **bias**, and finally pass it through an **activation function** which normalizes the output. This somewhat simple process is done over and over until finally our output layer has some *scores* or *predictions*.

Now to the learning.

These predictions are then compared to some **target**, or correct answer. We then use a **cost function** to determine how much we want to *punish* each of our guesses given how much they strayed from our target values. We then use this information to **backpropagate** across all our neurons and connections in order to adjust the biases and weights.

And this is it, *backpropagation* is how a neural network learns a particular task.

Details of the Visualization

At this point it may be worth to go ahead and play with the visualization a bit to see these components at work. What you'll notice is that I have given you the ability to adjust the inputs, each connection has the value of its weight hovering nearby, and each neuron has its bias (b) below and the result of its activation function σ above.

When you click `forward` you can see the final layers guesses in comparison to the target values. Then when you click `backprop` you can watch as the values are adjusted minutely. Then when you press `forward` again, you should see the the output layer improve slightly in comparison to the targets.

For those AI engineers out there, you'll notice there are certain complications we've ignored for the purpose of exposition. Right now, I use a softmax function to compute our cost, and a sigmoid function for activation. We have ignored other aspects of normal training like regularization, dropout, and mini-batching.

The trade-off being that, with less features to pay attention to, we can hone in on understanding the fundamentals involved in this procedure.

Interpreting Learning

With some background laid and the visualization explained, let's move on to talk about learning. The first thing to notice is, even in this simple network, paying attention to one particular number does not tell us much about how the entire system behaves. In fact, this is one of the reasons neural networks are hard to interpret: inspecting specific numbers in our system gives us little to no information about the overall dynamics.

So, when we think about learning, we don't want to get too invested in the meaning of tweaking each parameter. While this *is* the process, it's hard to see why this amounts to learning. Instead, we want to think about this in terms of **emergence**. That is, complex systems often run on simple rules, and those simple rules compound to create **emergent behaviors**. Conway's game of life is a great example of this, where we can see complicated structures form by just turning cells in a grid on and off according to a few basic rules.

Much like our brains, we cannot find our entire answer by looking at one particular region. Rather, we must try to make sense of how the different functional pieces give us the gestalt phenomenon of conscious thinking (which of course we are still struggling to understand).

Why we need to be careful to compare neural networks to brains is that brains have much more going on than these computational processes. Each time you train a neural network, you do so against a specific, singular task. Human learning, on the other hand, involves switching across contexts and redefining your task as you go along. At the physical level, our brains do not merely *adjust* their connections, but there chemical, electrical, and even quantum effects that determine how we rewire our brains and then act toward goals.

Having put these exceptions out in the open, we can still leverage the brain metaphor to understand our neural network's learning process. Feeding information forward, to us, is akin to receiving a new visual or tactile input and our brain processing this stimulus. Evaluating the cost function is the neural net's version of us evaluating a stimulus and determining the correct response. Finally, backpropagation is like the network reflecting on its errors so it can do better next time.

Given this learning mechanism, it's still unclear to us what kind of intelligence we'll see emerge from these systems in the coming years. However, it is important we all develop a realistic understanding in order to not over-aggrandize or under-anticipate what may be possible.

Hopefully using my visualization and this accompanying explanation we have a clearer picture of what "learning" really means. Getting our heads around the new systems we create is difficult, but also important for education, public communication, and choices around how to engineer systems with realistic expectations.