The AAAI-17 Workshop on
What's Next for AI in Games?
WS-17-15

Formalizing Anthropomorphism: A Study in Deep Neural Networks

Martin Zinkevich, Dale Schuurmans*
Google
1600 Amphitheatre Parkway
Mountain View, California 94043, USA
{martinz,schuurmans}@google.com

Abstract

Anthropomorphization can be used as a tool by humans to reason about complex nonhuman phenomena, by ascribing agents with intelligence and goals that are similar to their own. Deep neural networks are complex structures, and we do not understand well how to optimize them. One way to deal with such complexity is to give complex parts of the network (such as the activation functions) goals and actions, even if these parts are unchanging in their behavior. This allows us to transform the problem of finding the parameters of deep neural networks into a game, and use the same approaches that we use for games to generate good parameters for a deep neural networks. This paper presents the results of (Schuurmans and Zinkevich 2016) to the game theory community.

Introduction

Consider a mathematical programming problem

$$\text{Minimize } f(x) \tag{1}$$
$$\text{Subject to: } g(x) \leq 0 \ \forall g \in G \tag{2}$$
$$\text{and: } h(x) = 0 \ \forall h \in H \tag{3}$$

for a given objective $f : \mathbf{R}^n \to \mathbf{R}$ and finite sets $G, H \subseteq \mathbf{R}^{\mathbf{R}^n}$. We know how to solve such problems if they are linear or convex; for example, for unconstrained minimization, simple strategies such as calculating the gradient, performing a line search in the negative gradient direction, and repeating, is well known to work under general conditions, even if it is not always efficient. More sophisticated strategies, such as Newton's Method, or stochastic methods, such as stochastic gradient descent, Adagrad (Duchi, Hazan, and Singer 2011), SDCA (Shalev-Shwartz and Zhang 2013), et cetera, can do significantly better. However, all of these algorithms share a particular property: they converge in the limit to the correct answer.

When the objective is nonconvex, it is very difficult to guarantee convergence to a global minimum. Many algorithms (such as Adagrad (Duchi, Hazan, and Singer 2011)) used in training deep networks were developed to solve convex optimization problems. However, training a deep network is definitely not a convex optimization problem, so there are no guarantees that such algorithms will converge to a local minimum, let alone a global minimum. However, this assumption provides a way to generate reasonable algorithms for the problem. This is similar to how people optimize support vector machines (Boser, Guyon, and Vapnik 1992; Cortes and Vapnik 1995). For support vector machines, there are examples $x_1 \dots x_T \in \mathbf{R}^n$, and labels $y_1 \dots y_T \in \{-1, +1\}$, and we are trying to choose $\theta \in \mathbf{R}^n$ to minimize:

$$L(\theta) = \sum_{t=1}^{T} l(y_t, \theta \cdot x_t) \tag{4}$$

Theoretically, we want to define $l(y, \hat{y})$ to be a 0-1 margin loss, where $l(y, \hat{y}) = I(\lambda - y\hat{y} > 0)$, where $I(true) = 1$ and $I(false) = 0$, i.e. to minimize the number of examples where the prediction has the wrong (or nearly the wrong) sign. In practice, we set $l(y, \hat{y}) = \max(\lambda - y\hat{y}, 0)$ or some similar convex function. this assumption replaces an objective where theoretical guarantees hold on 0-1 loss on test data with objective that can be easily minimized via convex optimization. This has spurred amazing progress in the field of convex optimization, focused on this specific problem (Duchi et al. 2008; 2010; Shalev-Shwartz et al. 2011; Duchi, Hazan, and Singer 2011; Mukherjee et al. 2013; Zinkevich et al. 2010; Recht et al. 2011).

We believe that through deep learning and game theory, we can spur progress in the field of computational game theory, focused on trying to train deep networks.

Deep Neural Networks

Training a deep neural network consists of finding the parameters of the model that minimize loss. Unlike before, there is no simple assumption that can reduce this to a problem that can be reasoned about theoretically. At present, dropout (Srivastava et al. 2014), batch normalization (Ioffe and Szegedy 2015), gradient clipping (Pascanu, Mikolov, and Bengio 2013), Adagrad (Duchi, Hazan, and Singer 2011), regularization, and constraints on weights are all applied. As a key point, there is often no separation between the problem being solved and the method to solve it: many of these techniques blend the problem and the solution.

Outside of neural networks, the most classic example of this is decision trees. Theoretically, a decision tree could represent any binary function on the inputs: the generalization

*Also: University of Alberta, daes@ualberta.ca.

power comes from the weakness[1] of the learning algorithm, instead of a weakness in the space of models to be learned. While this nuance is powerful, it makes it difficult to evolve new algorithms in a non-experimental framework.

In (Schuurmans and Zinkevich 2016), we introduced the concept of Deep Games, which formalizes the notion of what it means to think of a deep neural network as a linear or convex problem. In a deep game, one learns a parameterized model $M : \mathbf{R}^n \to \mathbf{R}$. The structure can vary, but to be concrete, one can consider an example of a three layer, fully connected neural network with a $n_1 - n_2 - n_3$ structure (where $n_1 = n$ and $n_3 = 1$):

$$M(x) = A_3(\Theta_3^T A_2(\Theta_2 A_1(\Theta_1 x))) \tag{5}$$

where $\Theta_i \in \mathbf{R}^{n_i \times n_{i+1}}$, $A_i : \mathbf{R}^{n_i} \to \mathbf{R}^{n_i}$ is the activation function $a : \mathbf{R} \to \mathbf{R}$ applied to a vector such that for all $x \in \mathbf{R}^{n_i}$, $(A_i(x))_j = a(x_j)$.

The standard approach to minimizing loss over a parameterized neural network has been backpropagation, which is simply applying gradient descent or stochastic gradient descent to the problem. In and of itself, there is a limited understanding of gradient descent on non-convex problems. In particular, one can argue that it converges to a local minimum, given it starts reasonably close to it (Lee et al. 2016), but such an analysis has not yet been applied any of the more recent innovations on learning deep networks.

What makes deep networks hard are the activation functions. If a was an affine function, then we could find the optimal parameters by setting Θ_1 and Θ_2 to be the identity, and then tuning Θ_3, effectively making the whole problem convex. Yet, when we take the gradient of a at the current point, we are effectively assuming the activation function at each point is affine, but somehow changing over time in unpredictable ways. Basically, when we are applying odd variants of gradient descent to a nonconvex problem, we are anthropomorphizing parts of the loss function, considering their local behavior, but abandoning our ability to model their full behavior.

Anthropomorphizing Deep Learning

As shown in (Schuurmans and Zinkevich 2016), one can instead formalize the problem of training a deep neural network by thinking about agents. First, imagine that there is are three protagonists in charge of choosing Θ_1, Θ_2, and Θ_3. Now, in a learning problem, we have a bunch of examples, $\{(x_t, y_t)\}_{t \in \{1 \dots m\}}$. We want to minimize the loss:

$$L(\Theta_1, \Theta_2, \Theta_3) = \sum_{t=1}^{m} l(y_t, M(x_t)), \tag{6}$$

where l is a loss function that is convex in its second parameter. We could imagine the utility of the protagonists is opposite the loss: however, for even one protagonist to calculate her best response becomes difficult.

[1]Here, weakness refers to the ability of the algorithm to fit the training data perfectly. In machine learning, if you have perfectly fit the training data, you have likely "overfit", and will not do as well on new training data as if you had only loosely fit the training data.

We can imagine that the activation function "changes" with each example. Specifically, for every $t \in \{1 \dots m\}$, for every $i \in \{1, 2, 3\}$, there is a function $Z_{i,t} : \mathbf{R}^{n_i} \to \mathbf{R}^{n_i}$ such that $M_t : \mathbf{R}^n \to \mathbf{R}$ is defined as:

$$M_t(x) = Z_{3,t}(\Theta_3^T Z_{2,t}(\Theta_2 Z_{1,t}(\Theta_1 x))) \tag{7}$$

In particular, to model the activations we introduce an agent, which we refer to as a "zanni", to each layer in the neural network. We imagine Zanni[2] 1 chooses $Z_{1,t}$ for all $t \in \{1 \dots m\}$, Zanni 2 chooses $Z_{2,t}$ for all $t \in \{1 \dots m\}$ and Zanni 3 chooses $Z_{3,t}$ for all $t \in \{1 \dots m\}$. Now, in this anthropomorphization, it is important that what we observe matches our understanding of these anthropomorphized functions. Specifically, we want to design the game such that:

$$M_t(x_t) = M(x_t) \tag{8}$$

$$\nabla_\Theta M_t(x_t) = \nabla_\Theta M(x_t) \tag{9}$$

If $Z_{i,t}$ are first-order approximations of A at the right points, then M_t is a first-order approximation of M at x_t. Specifically, given the actions of the protagonists choosing Θ_1, Θ_2, Θ_3, the zanni agents must want and be able to choose $Z_{i,t}$ such that it is a first-order approximation of A: specifically, if it not the case, then at least one zanni can unilaterally deviate so that he gets more reward. Thus, at any Nash equilibrium, M_t will be a first-order approximation of M at x_t. However, this must be a choice: the mechanism by which the value and the gradient of $Z_{i,t}$ change must be an action of an agent: otherwise, a "unilateral" deviation by a protagonist has to take into consideration how the zannis respond, which is a computationally difficult task.

The action that the zanni has is to choose an affine $Z_{1,t}$ (or $Z_{2,t}$ or $Z_{3,t}$) given t. Define $Q_{1,t}, Q_{2,t}, Q_{3,t} \in \mathbf{R}^{n \times n}$, and $d_{1,t}, d_{2,t} \in \mathbf{R}^n$, and $d_{3,t} \in \mathbf{R}$, such that for all $i \in \{1, 2, 3\}$, $Z_{i,t}(x) = Q_{i,t}x + d_{i,t}$. We can write M_t as:

$$M_t(x_t) = Q_{3,t}(\Theta_3(Q_{2,t}(\Theta_2(Q_{1,t}(\Theta_1 x_t) + d_{1,t})) + d_{2,t})) + d_{3,t} \tag{10}$$

Now, $M_t(x_t)$ is a linear function of Θ_1 or Θ_2 or Θ_3. Our new utility function for the protagonists is:

$$u(\Theta_1, \Theta_2, \Theta_3, \{Q_{i,t}\}_{i,t}, \{d_{i,t}\}_{i,t}) = -\sum_{t=1}^{m} l(y_t, M_t(x_t)) \tag{11}$$

Define $T_1(x) = \Theta_1 x$, $T_2(x) = \Theta_2 A_1(T_1(x))$, and $T_3(x) = \Theta_3^T A_2(T_2(x))$. For $i \in \{1, 2, 3\}$, define $Q^*_{i,t} = \nabla A_i(v)|_{v = T_i(x_t)}$, and $d^*_{i,t} = A_i(T_i(x_t)) - Q^*_{i,t} T_i(x_t)$, such that $Q^*_{i,t} v + d^*_{1,t}$ is a first-order approximation of A_i at $T_i(x)$. Now, we define the utility of the ith zanni as $-\sum_{t=1}^{m} (Q_{i,t} - Q^*_{i,t})^2 + (d^*_{i,t} - d_{i,t})^2$. Thus, for all $i \in \{1, 2, 3\}$, the utility of Zanni i is maximized when $Q_{1,t} = Q^*_{1,t}$ and $d^*_{1,t} = d_{1,t}$.

[2]To complement the protagonist name from theatre, we chose to name a third set of agents in the game after the Zanni from Commedia dell'arte, who are both servants and tricksters, which is an apt characterization of the activation functions in deep networks.

Thus, we do not force the zannis to chose these first-order approximations: we merely make it in their best interest. This anthropomorphization both allows us to reasonably expect the zanni agents to play these first-order approximations, however does not force the protagonists choosing the parameters to contemplate how the zannis would change their behavior if the protagonists behavior changed. Thus, at any point in time, when a protagonist chooses a set of parameters, she sees a simple relationship between her choice of parameters and her utility, if she assumes that all the other protagonists and the zannis are fixed. Thus, we are capable of calculating the best response for any protagonist given the other agents are fixed, and moreover, are capable of minimizing regret. Moreover, we can prove that local and global minima of the deep neural network problem are Nash equilibria of our game. Another treatment of these details is fully worked out and explained in (Schuurmans and Zinkevich 2016).

Benefits

Two abilities determine the value of a new way of looking at an old problem: its ability to produce new solutions, and its ability to reason about which solutions might be successful. Anthropomorphization, in general, can justify almost any physical system: for instance, imagining that the north wind has a mind can explain in hindsight any weather, but only meteorology can predict the weather. Thus, it is the mathematical relationship between minima (more specifically, KKT points[3]) and Nash equilibria that makes this connection meaningful in our case. Also, practically, it is important to have a different view than the original problem, otherwise no additional creative stimulation is gained. For example, thinking solely about the individual layers as different agents does not simplify the problem enough to make it tractable.

An example where this view has spurred novel outcomes is the application of regret matching to training deep neural networks. In (Schuurmans and Zinkevich 2016), we provide multiple results that show how regret matching provides a viable, and often times advantageous training strategy for deep neural networks.

However, the model does not simply suggest any algorithm. For example, algorithms that deal with each layer separately but do not minimize regret, or an unconstrained variant of the problem with no limits on the weights or regularization are questionable in this environment.

Outlook

A final point about the perspective we are adopting in this work is an open question with regards to convergence: specifically, there is a well known result that internal regret minimizing algorithms converge to a correlated equilibrium in finite action games (Blum and Mansour 2007). Moreover, for some potential games, any correlated equilibrium is a mixture of pure strategy Nash equilibria (Neyman 1997). Now, there is a resemblance between the deep learning games and potential games: if you ignore the zannis, then the other agents all have the same equilibria.

Conjecture 1 *Given a game with agents P and agents Z, where agents P all have the same utility and minimize internal regret, and agents Z all play a unique best response every iteration, then is the limit of the joint action distribution a mixture of Nash equilibria?*

Notice that one can actually unify all the zannis here into one agent, if that simplifies the analysis.

Such a result would be remarkable, as it would in one sweep prove convergence for an entire set of algorithms. It would not prove convergence to a global minimum, since this is an NP hard problem (Blum and Rivest 1992). However, what it would do is allow for a variety of algorithms that trade off the speed of choosing a strategy in a single iteration and the number of iterations required to reach a certain value of internal regret. A second question would revolve around standard external regret minimizing algorithms, and how they would act in this environment: would they too converge to a mixture of Nash equilibria?

[3] A KKT point is a point where the Karush-Kuhn-Tucker conditions (Karush 1939; Kuhn and Tucker 1951) hold: if the gradient of the loss is zero, it is a KTT point, or roughly if the opposite of the gradient is exiting the feasible set. Under reasonable assumptions, every local minimum is a KKT point, but not every KKT point is a local minimum.

References

Blum, A., and Mansour, Y. 2007. From external to internal regret. *Journal of Machine Learning Research* 8:1307–1324.

Blum, A., and Rivest, R. 1992. Training a 3-node neural network is NP-complete. *Neural Networks* 5:117–127.

Boser, B. E.; Guyon, I. M.; and Vapnik, V. N. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth Annual Workshop on Computational learning theory*, 144.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine Learning* 20(3):273–297.

Duchi, J.; Shalev-Shwartz, S.; Singer, Y.; and Chandra, T. 2008. Efficient projections onto the l_1-ball for learning in high dimensions. In *Inter. Conf. on Machine Learning*, 272–279.

Duchi, J.; Shalev-Shwartz, S.; Singer, Y.; and Tewari, A. 2010. Composite objective mirror descent. In *COLT 2010 - The 23rd Conference on Learning Theory*, 14–26.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12:2121–2159.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR* abs/1502.03167.

Karush, W. 1939. Minima of functions of several variables with inequalities as side constraints. Master's thesis, Univ. of Chicago, Chicago, Illinois.

Kuhn, H., and Tucker, A. 1951. Nonlinear programming. In *Proceedings of 2nd Berkeley Symposium*, 481–492. University of California Press.

Lee, J.; Simchowitz, M.; Jordan, M.; and Recht, B. 2016. Gradient descent only converges to minimizers. In *29th Annual Conference on Learning Theory*.

Mukherjee, I.; Canini, K. R.; Frongillo, R. M.; and Singer, Y. 2013. Parallel boosting with momentum. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III*, 17–32.

Neyman, A. 1997. Correlated equilibrium and potential games. *International Journal of Game Theory* 26:223–227.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. *JMLR Workshop and Conference Proceedings* 28(3):1310–1318.

Recht, B.; Re, C.; Wright, S.; and Niu, F. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems 24*.

Schuurmans, D., and Zinkevich, M. 2016. Deep learning games. In *Advances in Neural Information Processing Systems*.

Shalev-Shwartz, S., and Zhang, T. 2013. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research* 14:567–599.

Shalev-Shwartz, S.; Singer, Y.; Srebro, N.; and Cotter, A. 2011. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming* 127(1):3–30.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Zinkevich, M.; Weimer, M.; Li, L.; and Smola, A. 2010. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems 23*.