The Difficulty of Learning Long-Term Dependencies with Gradient Flow in Recurrent Nets

Naresh Babu Bynagari

Director of Sales, Career Soft Solutions Inc, 145 Talmadge rd Edison NJ 08817, Middlesex, **USA**

*Corresponding Contact:
Email: naresh@careersoftusa.com

ABSTRACT

In theory, recurrent networks (RN) can leverage their feedback connections to store activations as representations of recent input events. The most extensively used methods for learning what to put in short-term memory, on the other hand, take far too long to be practicable or do not work at all, especially when the time lags between inputs and instructor signals are long. They do not provide significant practical advantages over, the backdrop in feedforward networks with limited time windows, despite being theoretically fascinating. The goal of this article is to have a succinct overview of this rapidly evolving topic, with a focus on recent advancements. Also, we examine the asymptotic behavior of error gradients as a function of time lags to provide a hypothetical treatment of this topic. The methodology adopted in the study was to review some scholarly research papers on the subject matter to address the difficulty of learning long-term dependencies with gradient flow in recurrent nets. RNNs are the most general and powerful sequence learning algorithm currently available. Unlike Hidden Markov Models (HMMs), which have proven to be the most successful technique in a variety of sequence processing applications, they are not limited to discrete internal states and can represent continuous, dispersed sequences. As a result, they can address problems that no other method can. Conventional RNNs, on the other hand, are difficult to train due to the problem of vanishing gradients.

Key words:
Recurrent Networks, Back-Propagation through Time, Learning Long-Term Dependencies, Gradient Flow

12/22/2020 **Source of Support:** None, **No Conflict of Interest:** Declared

INTRODUCTION

In theory, recurrent networks (RN) can leverage their better robustness to keep models of current contribution events in the guise of detection or activations. The most extensively used methods for learning what to put in short-term retention, on the other hand, take far too long to be practicable or do not work at all, notably when the latency lags among inputs and instructor signals are long. They do not provide significant practical advantages over effective tools in feedback systems with short time frames, despite being conceptually intriguing. “Back-Propagation Through Time" (BPTT) (Rumelhart, Hinton, and William, 1986; Schmidhuber, 1992; Bynagari, 2017, Figure 1) or “Real-Time Recurrent Learning" (RTRL) (Robinson and Fallside, 1987; Ganapathy, 2016) error signals owing back through time" appear to either (1) blow up or (2) fade away with traditional methods based on the comparison of the comprehensive gradient: the time-based evolution of the backpropagated error exponentially depends on the size of the weights (Bengio, Simard, and Frasconi, 1994; Bynagari, 2019). In scenario (1), fluctuating loads may result, whereas in case (2), learning to connect long temporal delays takes a while or may not operate at all.

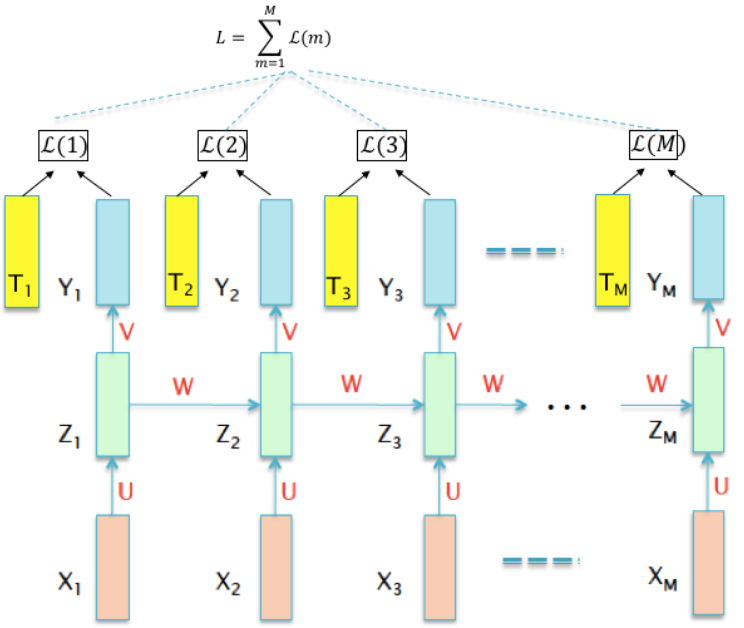

Figure 1: Back-Propagation through Time algorithm

Objective of the study

Because of several excellent breakthroughs that these algorithms have gained in the previous decade, the areas of Deep Learning (DL) and Neural Networks (NN) are generating a lot of buzzes. The goal of this article is to have a succinct overview of this rapidly evolving topic, with a focus on recent advancements (Bynagari, 2018). Also, we examine the asymptotic behavior of error gradients as a function of time lags to provide a hypothetical treatment of this topic.

This paper is divided into 5 sections, Section 1 is the introduction and objectives of the study. In Section 2 (review of related), we look at standard RNNs and use the approach first

proposed in Bynagari (2019) report to obtain the primary result. Also, in Section 2, we look at adaptive dynamical systems in general, which includes, in addition to standard RNNs, alternative recurrent designs based on different linkages and activation function choices (e.g., RBF or second-order connections). The analysis reported by (Bengio, Simard, and Frasconi, 1994) will be used to describe the two detrimental conditions that essentially arise; either the method is incompetent to strongly store past data about its feedbacks, or gradients fading exponentially. Section 3 (Methodology) will be described in this section. Lastly, we briefly explore other optimization strategies and designs that have been proposed to improve learning in the face of long-term dependencies in Section 4 (Results and discussion), and the last section is Section 5 (conclusion and recommendation).

REVIEW OF RELATED LITERATURE

Exponential error decay

Gradients of the error function

The outcomes attempted to prove to stand irrespective of the specific form or type of cost function applied (as long as it is nonstop in the throughput) and irrespective of the particular classifier which is used to analyze the gradient. Under gradients of the error function, explanation on how the typical "Back-Propagation through Time" classifier calculates gradients in a few words (Williams and Zipser, 1992; Vadlamudi, 2019; Bynagari & Fadziso, 2018).

The error at the time, t is signified by $E(T)$ considering only the error at the time, t, throughput unit k's error signal is $\delta_k\ (t) = \frac{\partial E\ (t)}{\partial net_k\ (t)}$ and some non-throughput unit $j's$ back-propagated error signal at time $r < t$ is $\delta_j\ (\tau) = f_j^t\left(net_j(\tau)\right)\left(\sum_i w_{ij}\ \delta_i\ (\tau + 1)\right)$, where $net_i\ (\tau) = \sum_i w_{ij} a_j(\tau - 1)$ is the unit $i's$ current net feedback, $a_i\ (\tau) = f_i(net_i\ (\tau))$ is the initiation or activation of non-feedback unit i with differentiable transmission function f_i and w_{ij} is the load on the link from j unit to i. The equivalent input to $w_{ij}'s$ entire load update is $\eta\delta_j\ (\tau)a_i(\tau - 1)$, where η signifies the learning proportion is, and 1 denotes a random unit linked to unit j.

Error Path integral

Consider a fully connected network with non-feedback unit directories ranging from 1 to n. concentrating on the local error drift from throughput unit k to random unit v (we'll show later that the analysis extends to global error ow as well). The error at k at time step t is distributed "back in time" for t - s time phases to a random unit v at time $s < t$ (Bynagari, 2019). The following fact is used to scale the error:

$$\frac{\partial \delta_v\ (s)}{\partial \delta_k\ (t)} = \begin{cases} f_v^t\ (net_v\ (t-1)\ w_{kv} & t - s = 1 \\ f_v^t\left(net_v\ (s)\right)\left(\sum_{i=1}^{n} \frac{\partial \delta_j\ (s+1)}{\partial \delta_k\ (t)}\ w_{kv}\right) & t - s > 1 \end{cases}$$

To answer the following problem, we shall unroll it across time and expand it (as completed for example in developing "Back-Propagation through Time"). In specific, let l be the index of a generic non-feedback unit in the network replica at the time for s < r < t, additionally, $l_s = v$ and $l_t = k$ We get the following results:

$$\frac{\partial \delta_v(s)}{\partial \delta_k(t)} = \sum_{l_{i-1}=1}^{n} \ldots \sum_{l_{i-1}=1}^{n} (w_{l_t l_t} (\prod_{\tau=t-1}^{s+1} f_{l_\tau}^I \left(net_{l_\tau}(\tau)\right) w_{l_\tau l_{\tau-1}}) f_{l_s}^I \left(net_{l_s}(s)\right)$$

Intuitive description of the above equation,

If $\left| f_{l_\tau}^I \left(net_{l_\tau}(\tau)\right) w_{l_\tau l_{\tau-1}} \right| > 1.0$ for all τ for all the leading product upsurges exponentially with t – s - 1. Which is, the error blows up, and contradictory error indications arriving at v unit can tip to fluctuating loads and unbalanced learning (for error blow-ups or divergences) (Pineda, 1988, Baldi and Pineda, 1991, Doya, 1992).

However, if $\left| f_{l_\tau}^I \left(net_{l_\tau}(\tau)\right) w_{l_\tau l_{\tau-1}}) \right| < 1.0$ < 1:0 for all τ, then the biggest product drops exponentially with t - s - 1. This is, the error disappears, and not anything can be learned in a suitable time. If f_l is the "logistic sigmoid function", then the utmost value of $f_{l_\tau}^I$ is constant and is greater than zero, then the magnitude of the gradient $\left| f_{l_\tau}^I \left(net_{l_\tau}(\tau)\right) w_{l_\tau l_{\tau-1}} \right|$ takes on utmost values where $w_{l_\tau l_{\tau-1}} = \frac{1}{a_{l_{\tau-1}}} \coth(\frac{1}{2} net_{l_\tau})$, the magnitude of the derivative approaches zero for $\left| w_{l_\tau l_{\tau-1}} \right| \to \infty$ and it less than 1.0 for $\left| w_{l_\tau l_{\tau-1}} \right| < 4.0$, for instance, if the absolute utmost load value w_{max} is less than 4.0). Henceforth with "conventional logistic sigmoid transfer functions" the error flow inclines to disappear as long as the loads have unconditional values less than 4.0, particularly at the start of the working out phase. The bigger initial loads do not support in overall as shown above for $\left| w_{l_\tau l_{\tau-1}} \right| \to \infty$, the appropriate derivative approaches zero "more rapidly" than the complete load can propagate (also, some loads may have to adjust their signs by intersection zero). Growing the learning proportion does not support either - the ratio of long-term error flow to short-term error flow leftovers untouched (Ganapathy, 2019a). "Back-Propagation through Time" is excessively prompted by current interruptions. It is worth noting that the summation term in the considered equation may have diverse signs as the number of units n upturns and does not always lead to an increase in error flow (Ganapathy, 2019b).

Weak upper limit for scaling factor

The succeeding, marginally extended disappearing error assessment also takes n, the amount of units, into consideration. For $t - s > 1$, this equation:

$$\frac{\partial \delta_v(s)}{\partial \delta_k(t)} = \sum_{l_{i-1}=1}^{n} \ldots \sum_{l_{i-1}=1}^{n} (w_{l_t l_t} (\prod_{\tau=t-1}^{s+1} f_{l_\tau}^I \left(net_{l_\tau}(\tau)\right) w_{l_\tau l_{\tau-1}}) f_{l_s}^I \left(net_{l_s}(s)\right)$$

Can be modified as:

$$(W_k^T).^T \; F^l(t-1)(\prod_{\tau=t-2}^{s+1} W \;\; F^l(\tau)) \; W_v f_v^I \left(net_v(s)\right)$$

Where the load matrix W is distinct by $\lfloor W \rfloor_{.ij} := w_{ij}$, $v's$ outbound load vector W_v is distinct by $\lfloor W_v \rfloor_{.i} := \lfloor W \rfloor_{.iv} = w_{iv}$, $k's$ inbound load vector W_k^T is distinct by $\lfloor W_k^T \rfloor_{.i} := \lfloor W \rfloor_{.ki} = w_{iv}$, and $F^l(t)$ is the diagonal matrix of 1st order derivatives distinct as: $\lfloor F^l(t) \rfloor_{.ij} := 0 \; if \; i \neq j, and \; \lfloor F^l(t) \rfloor_{.ij} := f_i^l\left(net_i(t)\right)$ if not. Here T is the inversion operator, $\lfloor A \rfloor_{.ij}$ are the component in the *i-th* column and *j-th* row of matrix A, and $\lfloor x \rfloor_{.i}$ is the *i-th* element of vector x.

Using a matrix model $\|.\|_{.A}$ compatible with vector model $\|.\|_{.x}$ to distinct:

$$f'_{max} := \max_{\tau=t-1,\ldots,s}\{\|F'(\tau)\|_A\}.$$

For $\max_{i=1,\ldots,n}\{|x_i|\} \leq \|x\|_x$ we get $\left|x^T y\right| \leq n \|x\|_x \|y\|_x$. Since

$$|f'_v(net_v(s))| \leq \|F'(s)\|_A \leq f'_{max},$$

we obtain the following inequality:

$$\left|\frac{\partial\delta_v(s)}{\partial\delta_k(t)}\right| \leq n\,(f'_{max})^{t-s}\,\|W_v\|_x\,\|W_k^T\|_x\,\|W\|_A^{t-s-2} \leq n\,(f'_{max}\,\|W\|_A)^{t-s}$$

$$f'_{max} := \max_{\tau=t-1,\ldots,s}\{\|F'(\tau)\|_A\}.$$

For $\max_{i=1,\ldots,n}\{|x_i|\} \leq \|x\|_x$ we get $\left|x^T y\right| \leq n \|x\|_x \|y\|_x$. Since

$$|f'_v(net_v(s))| \leq \|F'(s)\|_A \leq f'_{max},$$

we obtain the following inequality:

$$\left|\frac{\partial\delta_v(s)}{\partial\delta_k(t)}\right| \leq n\,(f'_{max})^{t-s}\,\|W_v\|_x\,\|W_k^T\|_x\,\|W\|_A^{t-s-2} \leq n\,(f'_{max}\,\|W\|_A)^{t-s}$$

This inequality results from

$$\|W_v\|_x = \|We_v\|_x \leq \|W\|_A\,\|e_v\|_x \leq \|W\|_A$$

and

$$\|W_k^T\|_x = \|e_k W\|_x \leq \|W\|_A\,\|e_k\|_x \leq \|W\|_A,$$

Where e_k is the vector unit is whose elements are zero except the *k-th* element, which is 1. Note that this is a weak, risky case upper limit – it will be stretched only if all $\|F^l(\tau)\|_{\cdot A}$ take greatest values, and if the inputs of all paths across which error flow back from unit k to unit v have the same sign. Big $\|W\|_{\cdot A}$ therefore, typically outcomes in small values of $\|F^l(\tau)\|_{\cdot A}$, as confirmed by experiments (Bynagari, 2019).

$$\|W\|_A := \max_r \sum_s |w_{rs}|$$

and

$$\|x\|_x := \max_r |x_r|,$$

we have $f'_{max} = 0.25$ for the logistic sigmoid. We observe that if

$$|w_{ij}| \leq w_{max} < \frac{4.0}{n} \quad \forall i,j,$$

then $\|W\|_A \leq n\ w_{max} < 4.0$ will result in exponential decay; by setting $\lambda := \left(\frac{n w_{max}}{4.0}\right) < 1.0$, we obtain

$$\left|\frac{\partial\delta_v(s)}{\partial\delta_k(t)}\right| \leq n\,\lambda^{t-s}.$$

$$\|W\|_A := \max_r \sum_s |w_{rs}|$$

and

$$\|x\|_x := \max_r |x_r|,$$

we have $f'_{max} = 0.25$ for the logistic sigmoid. We observe that if

$$|w_{ij}| \leq w_{max} < \frac{4.0}{n} \quad \forall i, j,$$

then $\|W\|_A \leq n \quad w_{max} < 4.0$ will result in exponential decay; by setting $\lambda := \left(\frac{n w_{max}}{4.0}\right) < 1.0$, we obtain

$$\left|\frac{\partial \delta_v(s)}{\partial \delta_k(t)}\right| \leq n \lambda^{t-s}.$$

$$\|W\|_A := \max_r \sum_s |w_{rs}|$$

and

$$\|x\|_x := \max_r |x_r|,$$

we have $f'_{max} = 0.25$ for the logistic sigmoid. We observe that if

$$|w_{ij}| \leq w_{max} < \frac{4.0}{n} \quad \forall i, j,$$

then $\|W\|_A \leq n \quad w_{max} < 4.0$ will result in exponential decay; by setting $\lambda := \left(\frac{n w_{max}}{4.0}\right) < 1.0$, we obtain

$$\left|\frac{\partial \delta_v(s)}{\partial \delta_k(t)}\right| \leq n \lambda^{t-s}.$$

Dilemma: Circumventing gradient decay averts long-term latching

The analysis of the problem of gradient decays is generalized to parameterized dynamical systems in Bengio et al. (1994), hence including second-order and other recurrent architectures. The fundamental theorem illustrates that obtaining gradient decay requires a sufficient condition, which is also a need for the system to reliably maintain discrete state data over time. To put it another way, when the loads and state trajectory are set up in such a way that the difficulty of gradient decay is obtained when a network may "latch" on data in its hidden units (i.e., depict long-term cravings). It's difficult to learn long-term dependencies when long-term gradients fade exponentially since this overall gradient is the sum of long-term and short-term influences, and the short-term influences entirely dominate the inclination (Vadlamudi, 2016).

These findings were based on a dissection of the state-space of the hidden layer into two types of territories: one in which gradients erode and one in which strong information latching is impossible. Let *y(t)* be the n-dimensional state vector at time *t* (for example, the vector [net$_1$(t),.., net (t)] when in view of a normal 1st-order recurrent network) and *y(t)* = *M(y(t - 1))* represent the map from the state at time *t - 1* to t for the independent deprived of a controller and dynamical structure (inputs) The disintegration described above is stated in terms of the $\lfloor M^l \rfloor > 1$ condition (no strong latching viable) or $\lfloor M^l \rfloor < 1$ (gradient disintegration), where $\lfloor M^l \rfloor$ is the model of the Jacobian (matrix of fractional results) of the map M. The focus of the investigation is on the lakes of attraction of M attractors in the domain of *y(t)* (or multiple within that space). The analysis focuses on this so exponential attractors, which are locally robust (but do not have to be fixed spots) and have M^l eigenvalues are greater than zero but less than 1 in absolute value. If a state (or a task of it) relics in a certain range, even in the presence of a given region of space (vs another region). When there are perturbations (such as noise in the inputs), it is likely to stock at least some of the data at least one byte of data for an undetermined amount of time (Bengio et al. 1994).

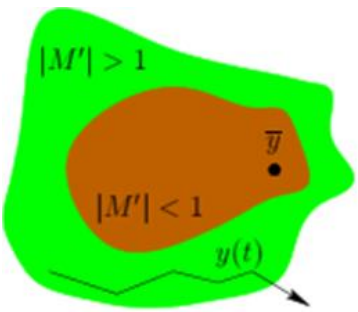

Figure 2: Robust latching (kick the formal out of a bowl of attraction)

It has been demonstrated that in regions where $\lfloor M^l \rfloor > 1$, arbitrarily tiny disturbances (for instance, due to inputs) can eventually kick the formal out of a bowl of attraction (Ortega and Rheinboldt, 1970) (as shown in the sample path in Figure 2). There is an equal of perturbation (liable on) lesser which the state will continue in the bowl of attraction (and) where $\lfloor M^l \rfloor < \lambda < 1$ there is a smooth of perturbation (liable on λ) below which the state will persist in the bowl of attraction as well as progressively tends to a definite volume about the attractor (see Figure 3). Due to this condition "information latching" subsequently it will permits to maintenance of separate data for the random intervals in the state variable *y(t).*

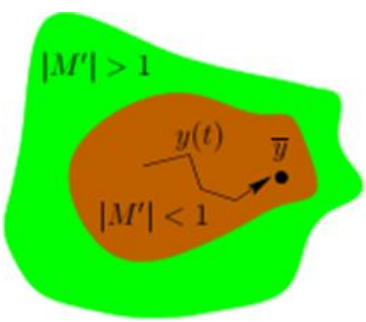

Figure 3: Robust latching showing persistent in the bowl of attraction as well as progressively tends to a definite volume about the attractor

Inappropriately, in the areas where $\lfloor M^l \rfloor < 1$ one can latch data and also express that gradient dwindling. The partial derivative of *y(t)* with respect to *y(s)* with $s < t$ is merely the products of the map derivatives among *s* and *t*:

$$\frac{\partial y\ (t)}{\partial y\ (s)} = \frac{\partial y(t)\ \partial y\ (t-1)}{\partial y\ (t-1)\partial y\ (t-2)} \cdots \frac{\partial y\ (s+1)}{\partial y\ (s)}$$

When the model of respective factors on the RHS is < 1, the LHS congregates exponentially firm to zero as $t - s$ rises. The consequence of this gradients decay can be expressed more explicitly as shown below:

$$\frac{\partial E\ (t)}{\partial W} = \sum_{\tau \le t} \frac{\partial E(t)\partial y(\tau)}{\partial y(\tau)\partial W} = \sum_{\tau \le t} \frac{\partial E(t)\partial y(t)\partial y(\tau)}{\partial y(t)\partial y(\tau)\partial W}$$

Therefore, for a span of the amount with $\tau \ll t_i$, that give rise

$$\left|\frac{\partial E(t)\partial y(\tau)}{\partial y(\tau)\partial W}\right| \to 0$$

In comparison to terms that are close to *t*, this term tends to turn comparatively small. This suggests that, notwithstanding the possibility of a variation in *W*, would allow $y(\tau\ )$ to jump to another (improved) attraction sink, the gradient That potential is not reflected in the cost with respect to W. The explanation is that a small modification in *W* would have a big impact in the not-too-distant past ($\tau\ close\ to\ t$) (Bengio et al. 1994).

METHODS

To achieve the objective of this study, we attempt to review some scholarly research papers on the subject matter to address the difficulty of learning long-term dependencies with gradient flow in recurrent nets. The analysis reported by (Bengio, Simard, and Frasconi, 1994) will take the central point of discussion and other related articles that will be of help to this discussion will be used to describe the two detrimental conditions that essentially arise; either the method is incompetent to strongly store past data about its feedbacks, or gradients fading exponentially. Also, we will briefly explore other optimization strategies and designs that have been proposed to improve learning in the face of long-term dependencies.

RESULTS AND DISCUSSION

Remedies

Gradient descent as a search strategy for finding peak loads in a recurrent neural network (RNN) has a fundamental drawback, as shown by the theoretical experiments above. To deal with the problem of long-term dependencies, several solutions have been made, some of which aim to tackle the optimization difficulty using alternate techniques. Others are working on alternate structures and search techniques (Ganapathy & Neogy, 2017). We will give you a quick rundown of these recommendations below.

Time Constants

Mozer (1992) employs time constants manipulating modifications in unit activations to deal with long-time lags ("deVries and Principe's related technique" (de Vries and Principe, 1991). maybe considered as mixing of "time-delay neural networks" (TDNN) (Lang, Waibel, and Hinton, 1990) and the passage of time variables). Therefore, for long time lapses,

external time constants are required fine turning (Mozer, 1992). The alternate approach proposed by Sun et al. (1993) adds the old activation and the (scaled) current to the activity of a recurrent unit input (net). The net input, on the other hand, tends to disturb the stored data, making long-term storage impossible. Lin et al. (1996) also offer NARX networks, which are time-delay network versions. Gradient ow can be enhanced in this design because embedded memory effectively introduces "shortcuts" in error propagation, a journey through time The same concept can be used in different types of architecture. Rather than using a single delay, many delays are used in the connections between hidden state units compared to output units (Lin et al., 1998). Nevertheless, these structural designs cannot resolve the general difficulty for the reason that they can only lengthen the interval of the temporal dependencies that can be learned by a continuous multiplicative factor. In conclusion, El Hihi and Bengio (1996) looked at recurrent neural networks that were arranged hierarchically. In view of varying levels of time coefficients or delays.

Ring's Technique

Ring also suggests an approach for bridging long-time gaps in his paper (Ring, 1993). He adds a higher-order unit manipulating appropriate links at whatever time a unit in his network receives contradictory error signals, that is, certain error signals advocate increasing the unit's activity while others suggest decreasing it. Even though his method can be exceedingly fast at times, bridging a hundred-step time lapse may require the adding of hundred units. Ring's net also doesn't put on to lag times that aren't visible.

Searching deprived of gradients

The large-scale optimization strategy that directs the search for a load solution is directly tied to the difficulty of learning long-term connections (Neogy & Bynagari, 2018). Other types of weight space searches, in which the algorithms for producing another candidate weight solution are not dependent on uninterrupted gradients, are one way to circumvent the difficulty (Ganapathy, 2018). The proposed algorithm, multi-grid random search, and discrete regression coefficient are among the strategies explored by Bengio et al. (1994) Angeline et al. (1994) suggest a genetic method that eliminates gradient computing as well.

The most basic type of search without gradient, on the other hand, just randomizes all network weights until the resulting net properly classifies all training sequences. In reality, simple weight guessing answers some common benchmarks mentioned in prior work quicker than the recurrent net techniques provided therein, as stated in simple weight guessing (Bynagari & Amin, 2019). This isn't to say that guessing weights is a good method. It just means that the issues are straightforward. More realistic tasks necessitate a large number of free parameters (e.g., input loads) or a high level of weight precision (for example; for continuous-valued factors), making guessing impossible. At the moment, it's unclear what's going on. In the case of more realistic problems, it is currently unknown to what extent more advanced gradient-less algorithms can outperform guessing.

Probabilistic aim propagation

For propagating aims, Bengio and Frasconi (1994) suggest a probabilistic technique. With n so-called "state networks," their device can be in one of only n state variables at any given time. The parameters are tweaked with the help of the implicit understanding algorithm is a method for maximizing expected value. However, in order to tackle issues that necessitate such systems require a significant amount of memory to retain contextual information would necessitate an insurmountable number of states (i.e., state networks).

Adaptive categorization chunkers

In theory, Schmidhuber's hierarchical chunker systems (Schmidhuber, 1992; Bynagari, 2017) can bridge arbitrary time gaps, but only if local predictable exists throughout the sub-sequences creating the time lags (Mozer, 1992). Schmidhuber, for example, employs hierarchical recurrent networks with conscience time scales to handle specific grammar learning tasks with minimal time lags in excess of one thousand steps (Bynagari, 2017). Conversely, when the noise level rises and the input categorizations develop less compressible, chunker systems' presentation suffers (Paruchuri, 2019).

Long Short-Term Memory

The "Long Short-Term Memory" (LSTM) method is a new, efficient, gradient-based approach. “The LSTM algorithm was created to solve the disappearing error problem. “Long Short-Term Memory" can learn to bridge small time gaps in excess of one thousand discrete time steps by trimming the gradient where it is not detrimental by using constant error carrousels to implement constant error flow inside the Special Forces”. Multiplicative gate units figure out how to open and close the gate, there is a continual error flow. In both time and space, the "Long Short-Term Memory" is local; its computational complication per time step and load is O. (1). Local, distributed, real-valued, and noisy pattern representations have all been used in artificial data experiments thus far. "Long Short-Term Memory" outperformed Back-Propagation through Time" (BPTT), “Real-Time Recurrent Learning" (RTRL), Recurrent Cascade-Correlation (RCC), Elman networks, and Neural Sequence Chunking in terms of a number of successful runs and speed of learning.

CONCLUSION

RNNs are the most general and powerful sequence learning algorithm currently available. Unlike Hidden Markov Models (HMMs), which have proven to be the most successful technique in a variety of sequence processing applications, they are not limited to discrete internal states and can represent continuous, dispersed sequences. As a result, they can address problems that no other method can. Conventional RNNs, on the other hand, are difficult to train due to the problem of vanishing gradients. We believe this is why feedforward neural networks have more successful real-world applications than RNNs. Some of the solutions presented in this chapter may result in more efficient learning systems. Long lime lag research, on the other hand, appears to be in its infancy; no commercial uses of any of these approaches have been reported thus far. Long temporal lags are problematic for any soft computing technology, including RNNs. When dealing with extended sequences (such as speech or biological data), HMMs often use a localized representation of time via highly limited non-ergodic transition diagrams (different states are designed for different portions of a sequence). Diffusion of credit [5,] a phenomenon that closely mimics the vanishing gradients problem in RNNs, does not efficiently propagate belief over extended time lags.

REFERENCES

Angeline, P. J., Saunders, G. M. and Pollack, J. P. (1994). An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks, 5*(1):54 - 65, 1994.

Bald, P. and Pineda, F. (1991). Contrastive learning and neural oscillator. *Neural Computation, 3*, 526 - 545.

Bengio, Y. and Frasconi, P. (1994). Credit assignment through time: Alternatives to backpropagation. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, Advances in Neural Information Processing Systems 6, pages 75{82. San Mateo, CA: Morgan Kaufmann, 1994.

Bynagari, N. B. (2017). Prediction of Human Population Responses to Toxic Compounds by a Collaborative Competition. *Asian Journal of Humanity, Art and Literature*, *4*(2), 147-156. https://doi.org/10.18034/ajhal.v4i2.577

Bynagari, N. B. (2018). On the ChEMBL Platform, a Large-scale Evaluation of Machine Learning Algorithms for Drug Target Prediction. *Asian Journal of Applied Science and Engineering*, *7*, 53–64. Retrieved from https://upright.pub/index.php/ajase/article/view/31

Bynagari, N. B. (2019). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Asian Journal of Applied Science and Engineering*, *8*, 25–34. Retrieved from https://upright.pub/index.php/ajase/article/view/32

Bynagari, N. B., & Amin, R. (2019). Information Acquisition Driven by Reinforcement in Non-Deterministic Environments. *American Journal of Trade and Policy*, *6*(3), 107-112. https://doi.org/10.18034/ajtp.v6i3.569

Bynagari, N. B., & Fadziso, T. (2018). Theoretical Approaches of Machine Learning to Schizophrenia. *Engineering International*, *6*(2), 155-168. https://doi.org/10.18034/ei.v6i2.568

de Vries, B. and Principe, J. C. (1991). A theory for neural networks with time delays. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, Advances in Neural Information Processing Systems 3, pages 162 - 168. San Mateo, CA: Morgan Kaufmann.

Doya, K. (1992). Bifurcations in the learning of recurrent neural networks. In Proceedings of 1992 IEEE International Symposium on Circuits and Systems, pp. 2777 - 2780.

Ganapathy, A. (2016). Virtual Reality and Augmented Reality Driven Real Estate World to Buy Properties. *Asian Journal of Humanity, Art and Literature*, *3*(2), 137-146. https://doi.org/10.18034/ajhal.v3i2.567

Ganapathy, A. (2018). Cascading Cache Layer in Content Management System. *Asian Business Review*, *8*(3), 177-182. https://doi.org/10.18034/abr.v8i3.542

Ganapathy, A. (2019a). Image Association to URLs across CMS Websites with Unique Watermark Signatures to Identify Who Owns the Camera. *American Journal of Trade and Policy*, *6*(3), 101-106. https://doi.org/10.18034/ajtp.v6i3.543

Ganapathy, A. (2019b). Mobile Remote Content Feed Editing in Content Management System. *Engineering International*, *7*(2), 85-94. https://doi.org/10.18034/ei.v7i2.545

Ganapathy, A., & Neogy, T. K. (2017). Artificial Intelligence Price Emulator: A Study on Cryptocurrency. *Global Disclosure of Economics and Business*, *6*(2), 115-122. https://doi.org/10.18034/gdeb.v6i2.558

Lin, T. Horne, B. G., Ti~no, P. and Giles, C. L. (1996). Learning long-term dependencies in NARX recurrent neural networks. IEEE Transactions on Neural Networks, 7(6):1329 - 1338, November 1996.

Lin, T., Horne, B. G. and Giles, C. L. (1998). How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies. Neural Networks, 11(5):861 – 868.

Mozer, M. C. (1992). Induction of multiscale temporal structure. In D. S. Lippman, J. E. Moody, and D. S. Touretzky, editors, Advances in Neural Information Processing Systems 4, pages 275 - 282. San Mateo, CA: Morgan Kaufmann.

Neogy, T. K., & Bynagari, N. B. (2018). Gradient Descent is a Technique for Learning to Learn. *Asian Journal of Humanity, Art and Literature*, *5*(2), 145-156. https://doi.org/10.18034/ajhal.v5i2.578

Ortega, J. M. and Rheinboldt, W.C. (1970). Iterative Solution of Non-linear Equations in Several Variables and Systems. Academic Press, New York.

Paruchuri, H. (2019). Market Segmentation, Targeting, and Positioning Using Machine Learning. *Asian Journal of Applied Science and Engineering*, *8*(1), 7-14.

Pineda, F. J. (1988). Dynamics and architecture for neural computation. Journal of Complexity, 4:216 - 245.

Ring, M. B. (1993). Learning sequential tasks by incrementally adding higher orders. In J. D. Cowan S. J. Hanson and C. L. Giles, editors, Advances in Neural Information Processing Systems 5, pages 115{122. Morgan Kaufmann.

Robinson, A. J. and Fallside, F. (1987). The utility-driven dynamic error propagation network. Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department, 1987.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning internal representations by error propagation. In Parallel Distributed Processing, volume 1, pages 318{362. MIT Press.

Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. Neural Computation, 4(2):234 - 242,

Vadlamudi, S. (2016). What Impact does Internet of Things have on Project Management in Project based Firms?. *Asian Business Review*, *6*(3), 179-186. https://doi.org/10.18034/abr.v6i3.520

Vadlamudi, S. (2019). How Artificial Intelligence Improves Agricultural Productivity and Sustainability: A Global Thematic Analysis. *Asia Pacific Journal of Energy and Environment*, *6*(2), 91-100. https://doi.org/10.18034/apjee.v6i2.542

Williams, R. J. and Zipser, D. (1992). Gradient-based learning algorithms for recurrent networks and their computational complexity. In Backpropagation: Theory, Architectures and Applications. Hillsdale, NJ: Erlbaum.

--0--