On the Difficulty of Training RNNs (ICML, 2013)

R. Pascanu, T. Mikolov, Y. Bengio

Poster presented by: Hanna M. Dettki, Anagha Radhakrishna Palandye, Juechen Zhong, Harshit Bhargava

1. Introduction & Background

1.1 Context & Motivation

- **Importance of sequence modeling**
  - e.g., language, time-series in finance
- Identifying gradient problems [1]
  - **Vanishing gradient problem**: impossible to learn long-term dependencies
  - **Exploding gradient problem**: numerical instabilities → unstable training
- → **a.** Stable gradient propagation as a prerequisite for learning long-range dependencies in RNNs **b.** Gradient norm clipping and temporal sensitivity regularization as targeted solutions (both **a** & **b** are paper's contributions)

1.2 Schematic & formal def. of a Recurrent Neural Network (RNN)

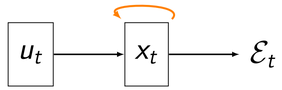

Fig. 1: **The recurrent connections** in the hidden layer allow information to persist from one input to another.

$$x_t = F(x_{t-1}, u_t, \theta) \quad (1) \text{ General}$$
$$x_t = W_{\text{rec}}\,\sigma(x_{t-1}) + W_{in}u_t + \mathbf{b} \quad (2) \text{ used in the paper}$$

where • u_t: input, • x_t: state, • t: time step, • $\mathbf{b}$: bias, • $\mathcal{E}_t = \mathcal{L}(x_t)$ (error), • W_{rec}: recurrent weight matrix

1.3 Training RNNs: Backprop Through Time (BPTT) on Unrolled RNN

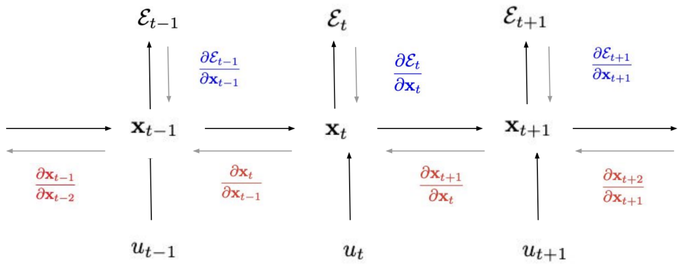

Fig. 2: Unrolled RNN: Creating a copy of the model for each time step. $\mathcal{E}_t$: error obtained at time step t from the output — • *total gradient over time* • *temporal error contribution*

$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{t=1}^{T} \frac{\partial \mathcal{E}_t}{\partial \theta} \quad (3)$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{k=1}^{t} \left(\frac{\partial \mathcal{E}_t}{\partial x_t} \frac{\partial x_t}{\partial x_k} \frac{\partial^+ x_k}{\partial \theta} \right) \quad (4)$$

$$\frac{\partial x_t}{\partial x_k} = \prod_{i=k+1}^{t} W_{\text{rec}}{}^{\top} \cdot \text{diag}\left(\sigma'(x_{i-1})\right) \quad (5)$$

where $\frac{\partial^+ x_k}{\partial \theta}$ denotes the "immediate" partial derivative (treating x_{k-1} as constant).

2. The Problem

2.1 Mechanics of Exploding and Vanishing Gradients:

- These issues occur in RNNs due to repeated multiplication of Jacobian matrices during backpropagation.
- The behavior of gradients depends on the spectral radius ρ of the recurrent weight matrix W_{rec}:
  - If $\rho < 1$, gradients **vanish**.
  - If $\rho > 1$, gradients **explode**.
- For non-linear activations with bounded derivatives (e.g., $\gamma = 1$ for $\tanh$), gradients **vanish** when the largest singular value $\lambda_1 < \gamma^{-1}$.

2. The Problem (cont.)

2.2 Dynamical Systems View:

- An RNN's hidden state evolves like a dynamical system converging to attractors.
- As parameters change, the system may cross bifurcation points, causing drastic changes in state evolution.
- Crossing basin boundaries can result in gradient explosions.
- Inputs can shift the system into different attractor basins, intensifying this instability.

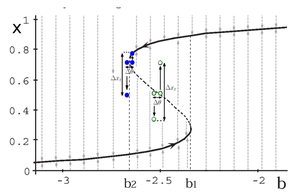

Fig.3:Bifurcation for attractor transitions

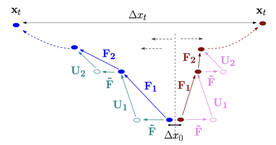

Fig.4:Grad explosion due to basin crossing

2.3 Geometric Interpretation:

- Consider $x_t = w\sigma(x_{t-1}) + b$ with $x_0 = 0.5$.
- In the linear case ($b = 0$), gradients are $\frac{\partial x_t}{\partial w} = tw^{t-1}x_0$, showing exponential growth.
- Exploding gradients align with steep directions in the error surface.
- These steep directions form sharp walls that SGD struggles to traverse, disrupting convergence.

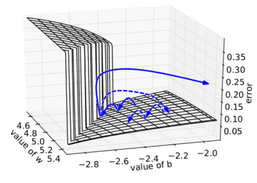

Fig.5:Steep error surface caused by exploding gradients

3. Solutions

3.1 Gradient Clipping

- Pseudo-code: $\hat{g} \leftarrow \nabla E$; if $\|\hat{g}\|_2 \geq \tau$ then $\hat{g} \leftarrow \tau \cdot \frac{\hat{g}}{\|\hat{g}\|_2}$
- Gradient clipping introduces a hyperparameter: the threshold. A common heuristic sets this value based on the average gradient norm over early training steps.
- Compared to clipping individual gradient components by value, norm-based clipping preserves the direction of the gradient vector and is generally more robust in high-dimensional settings.

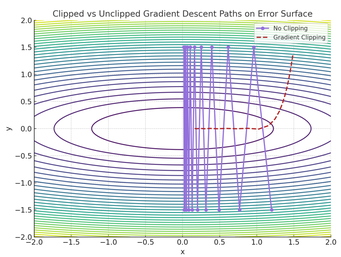

Fig.6:Clipped vs Unclipped Gradient Descent Path on Error Surface

3.2 Vanishing Gradient Regularization

- Regularizer:

$$\Omega = \sum_k \Omega_k = \sum_k \left(\frac{\left\| \frac{\partial \mathcal{E}}{\partial x_{k+1}} \cdot \frac{\partial x_{k+1}}{\partial x_k} \right\|}{\left\| \frac{\partial \mathcal{E}}{\partial x_{k+1}} \right\|} - 1 \right)^2 \quad (6)$$

$$\frac{\partial^+ \Omega}{\partial W_{\text{rec}}} = \sum_k \frac{\partial^+}{\partial W_{\text{rec}}} \left[\left(\frac{\left\| \frac{\partial \mathcal{E}}{\partial x_{k+1}} \cdot W_{\text{rec}}^{\top} \cdot \text{diag}\left(\sigma'(x_k)\right) \right\|^2}{\left\| \frac{\partial \mathcal{E}}{\partial x_{k+1}} \right\|^2} - 1 \right)^2 \right] \quad (7)$$

- The regularization term only enforces norm preservation of the Jacobian matrix $\frac{\partial x_{k+1}}{\partial x_k}$ in the direction of the error signal $\frac{\partial \mathcal{E}}{\partial x_{k+1}}$, not in all directions.
- The soft constraint does not guarantee perfect norm preservation, so exploding gradients may still occur, particularly during early training or unstable updates. To mitigate this, we combine the regularizer with gradient clipping for more stable and effective learning.

4. Experiments & Results

- **Datasets Utilized:** Synthetic pathological tasks (temporal order, addition, multiplication, 3-bit temporal order, random permutation, noiseless memorization)[2]. Natural datasets polyphonic music prediction [3](Piano-midi.de, Nottingham, MuseData) and character-level language modeling [4](Penn Treebank), testing both short and long-term dependency learning.
- **Temporal Order Problem Success:** Showed that gradient clipping (MSGD-C) and regularization (MSGD-CR) improved success rates over standard mini-batch SGD (MSGD), especially for longer sequences.
- **Initialization Impact:** 3 initializations: sigmoid ($W_{rec}, W_{in}, W_{out} \sim \mathcal{N}(0, 0.01)$), basic tanh ($W_{rec}, W_{in}, W_{out} \sim \mathcal{N}(0, 0.1)$), and smart tanh ($W_{in}, W_{out} \sim \mathcal{N}(0, 0.01)$, sparse W_{rec} with spectral radius 0.95). Smart tanh performed best.
- **Clipping Importance:** Gradient clipping was critical for tasks needing long memory traces, as longer sequences correlated with larger spectral radii, increasing the likelihood of exploding gradients.
- **Generalization Across Lengths:** A single model trained with MSGD-CR handled sequences from 50 to 200 steps with 100% success and generalized to unseen lengths up to 5000 steps, suggesting robust long-term memory.
- **Natural Problems Tested:** Applied to music prediction and language modeling, with clipping and regularization improving performance on real-world tasks.
- **Clipping as Optimization:** Clipping improved both training and test errors in natural tasks, indicating it addresses optimization issues rather than acting as a regularizer.
- **Regularization Challenges:** Fixed regularization weights harmed short-term correlation learning in natural tasks; a decreasing schedule (halving) was needed, suggesting a trade-off between short- and long-term dependencies.
- **SOTA Results:** Penn Treebank:MSGD-CR matched RNN SOTA results; Music prediction:achieved RNN SOTA, RNN-NADE performed better, validating the clipping strategy.

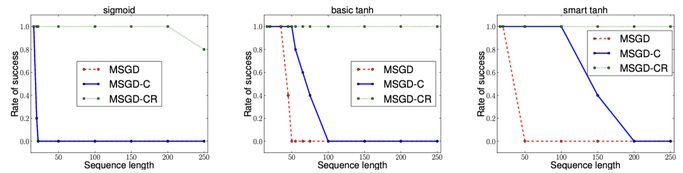

Fig.7:Rate of success-temporal order problem vs sequence length for different initializations

5. Relevance today & SOTA techniques

- **Exploding gradients: Clipping is still relevant!**
- **Vanishing gradients: More recent alternatives to regularization:**
  - Residual connections [5]
  - Layer normalization [8]
  - Attention mechanism [10,11]
  - Gating mechanisms [6,7]
  - Gradient checkpointing [9]
  - Positional encoding [10,12,13]

6. References

[1]Bengio,Y., Simard,P., and Frasconi,P. (1994). Learn- ing long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks , 5(2), 157–166.
[2]Hochreiter,S. and Schmidhuber,J. (1997).Long short-term memory.Neural Computation, 9(8), 1735–1780.
[3]Boulanger-Lewandowski,N., Bengio,Y., and Vincent,P. (2012). Modeling temporal dependencies in high- dimensional sequences: Application to polyphonic music generation and transcription.In Proceed- ings of the Twenty-nine International Conference on Machine Learning (ICML'12).ACM.
[4]Mikolov, T., Sutskever, I., Deoras, A., Le, H.-S., Kombrink, S. and Cernocky, J.(2012). Subword language modeling with neural networks.preprint (http://www.fit.vutbr.cz/imikolov/rnnlm/char.pdf).
[5] He, Kaiming, et al. "Deep residual learning for image recognition." the IEEE conference on computer vision and pattern recognition. 2016.
[6] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).
[7] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
[8] Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer normalization." arXiv preprint arXiv:1607.06450 (2016).
[9] Chen, Tianqi, et al. "Training deep nets with sublinear memory cost." arXiv preprint arXiv:1604.06174 (2016).
[10] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
[11] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
[12] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). 2019.
[13] Su, Jianlin, et al. "Roformer: Enhanced transformer with rotary position embedding." Neurocomputing 568 (2024): 127063.