On the Difficulty of Training RNNs (ICML, 2013)

R. Pascanu, T. Mikolov, Y. Bengio

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 (Bengio et al., 1994)
  - **Vanishing gradient problem**: impossible to learn long-term dependencies
  - **Exploding gradient problem**: numerical instabilities → unstable training
- → Why stable gradient flow is critical for learning temporal dependencies (paper's contribution)

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

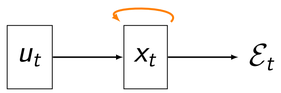


Fig. 1: **The recurrent connections** in the hidden layer allow information to persist from one input to another.

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

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

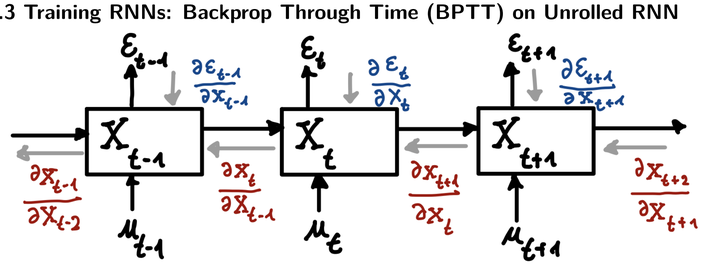


Fig. 2: Unrolled RNN: Creating a copy of the model for each time step. $\mathcal{E}_t$: error obtained at time step t from the output — • *total gradient over time* • *temporal error contribution*

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

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

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

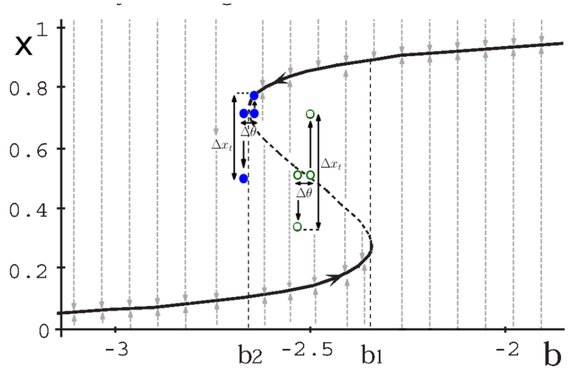
where $\frac{\partial^+ x_k}{\partial \theta}$ denotes the "immediate" partial derivative (treating x_{k-1} as constant).

2. The Problem

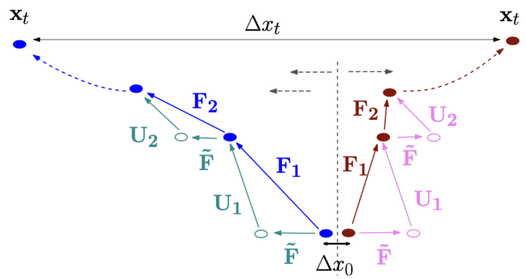
Mechanics of Exploding and Vanishing Gradients:
These issues occur in RNNs due to repeated multiplication of Jacobian matrices during backpropagation. If the spectral radius ρ of the recurrent weight matrix W_{rec} is less than 1, gradients vanish; if greater than 1, they explode. For non-linear activations with bounded derivatives (e.g., $\gamma = 1$ for tanh), gradients vanish when the largest singular value $\lambda_1 < \gamma^{-1}$.

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.

2. The Problem (cont.)

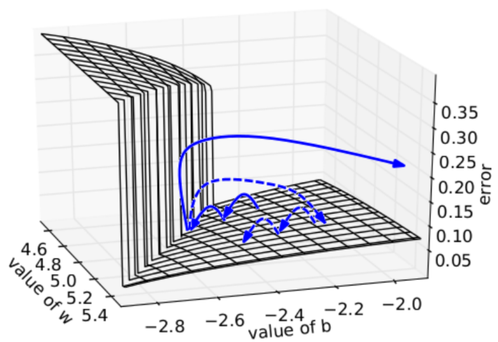


Bifurcation diagram showing attractor transitions



Gradient explosion due to basin crossing

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, forming sharp walls that SGD struggles to traverse, disrupting convergence.



Steep error surface caused by exploding gradients

5. Relevance today & SOTA techniques

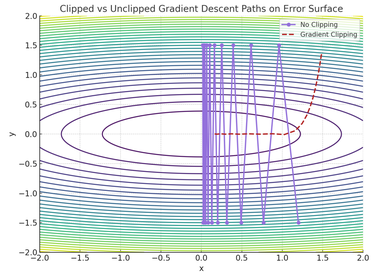
Exploding gradients: **Clipping is still relevant!**
Vanishing gradients: **More recent alternatives to regularization:**

- Residual connections • Gating mechanisms • Attention mechanism
- Gradient checkpointing • Layer normalization • Positional encoding

3. Solution

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.



Vanishing Gradient Regularization

- Regularizer:

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

$$\frac{\partial^+ \Omega}{\partial W_{\text{rec}}} = \Sigma_k \frac{\partial^+}{\partial W_{\text{rec}}} \left(\left(\frac{\left\| \frac{\partial E}{\partial x_{k+1}} \cdot W_{\text{rec}}^{\top} \cdot \operatorname{diag}(\sigma'(x_k)) \right\|^2}{\left\| \frac{\partial E}{\partial x_{k+1}} \right\|^2} - 1 \right)^2 \right)$$

- 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 $\frac{\partial E}{\partial x_{k+1}}$, 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: Experiments leveraged synthetic pathological tasks (temporal order, addition, multiplication, 3-bit temporal order, random permutation, noiseless memorization) (Hochreiter and Schmidhuber (1997)), and natural datasets including polyphonic music prediction (Piano-midi.de, Nottingham, MuseData) and character-level language modeling (Penn Treebank), testing both short and long-term dependency learning.
- Temporal Order Problem Success: The temporal order problem showed that gradient clipping (MSGD-C) and regularization (MSGD-CR) improved success rates over standard mini-batch SGD (MSGD), especially for longer sequences.
- Impact of Initialization: Three initializations (sigmoid, basic tanh, smart tanh) were compared, with "smart tanh" (sparse W_rec, spectral radius 0.95) performing best, highlighting initialization's role in RNN training.
- 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.
- Other Pathological Tasks: MSGD-CR solved multiple tasks (addition, multiplication, 3-bit temporal order, random permutation, noiseless memorization) with near-perfect success for sequences up to 200 steps, outperforming prior work.
- Natural Problems Tested: The solutions were applied to real-world tasks: polyphonic music prediction (Piano-midi.de, Nottingham, MuseData) and character-level language modeling (Penn Treebank).
- 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: For Penn Treebank, MSGD-CR matched RNN SOTA results; for music prediction, it achieved RNN state-of-the-art, though RNN-NADE performed better overall, validating the clipping strategy.

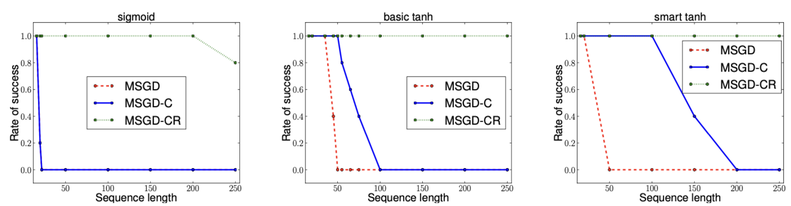


Figure 7: Rate of success for solving the temporal order problem versus sequence length for different initializations (from left to right: sigmoid, basic tanh and smart tanh)