Get started Open in app

towards
data science

Follow 575K Followers

HANDS-ON TUTORIALS

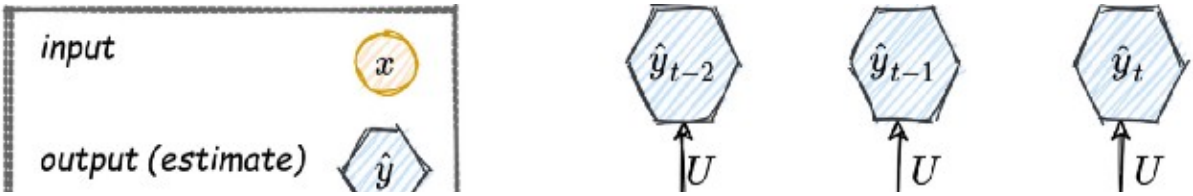
The Exploding and Vanishing Gradients Problem in Time Series

In this post, we deal with exploding and Vanishing Gradient in Time Series and in particular in Recurrent Neural Network (RNN) by Truncated BackPropagation Through Time and Gradient Clipping.

Barak Or Oct 10, 2020 · 6 min read

Intro

In this post, we focus on deep learning for sequential data techniques. All of us familiar with this kind of data. For example, the text is a sequence of words, video is a sequence of images. More challenging examples are from the branch of time series data, with medical information such as heart rate, blood pressure, etc., or finance, with stock price information. The most common *AI* approaches for time-series tasks with deep learning is the Recurrent Neural Networks (RNNs). The motivation to use RNN lies in the generalization of the solution with respect to time. As sequences have different lengths (mostly), a classical deep learning architecture such as Multy Layers Perceptrons (MLP) can not be applied without modifying it. Moreover, the number of weights in MLP is absolutely huge! Hence, The RNN is commonly used, where the weights are shared during the entire architecture. A simple RNN architecture is shown below, where V, W, and U are the weights matrices, and b is the bias vector.



Get started Open in app

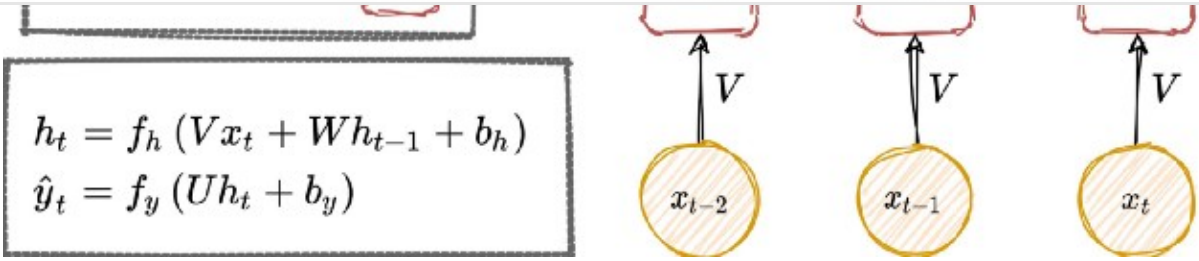


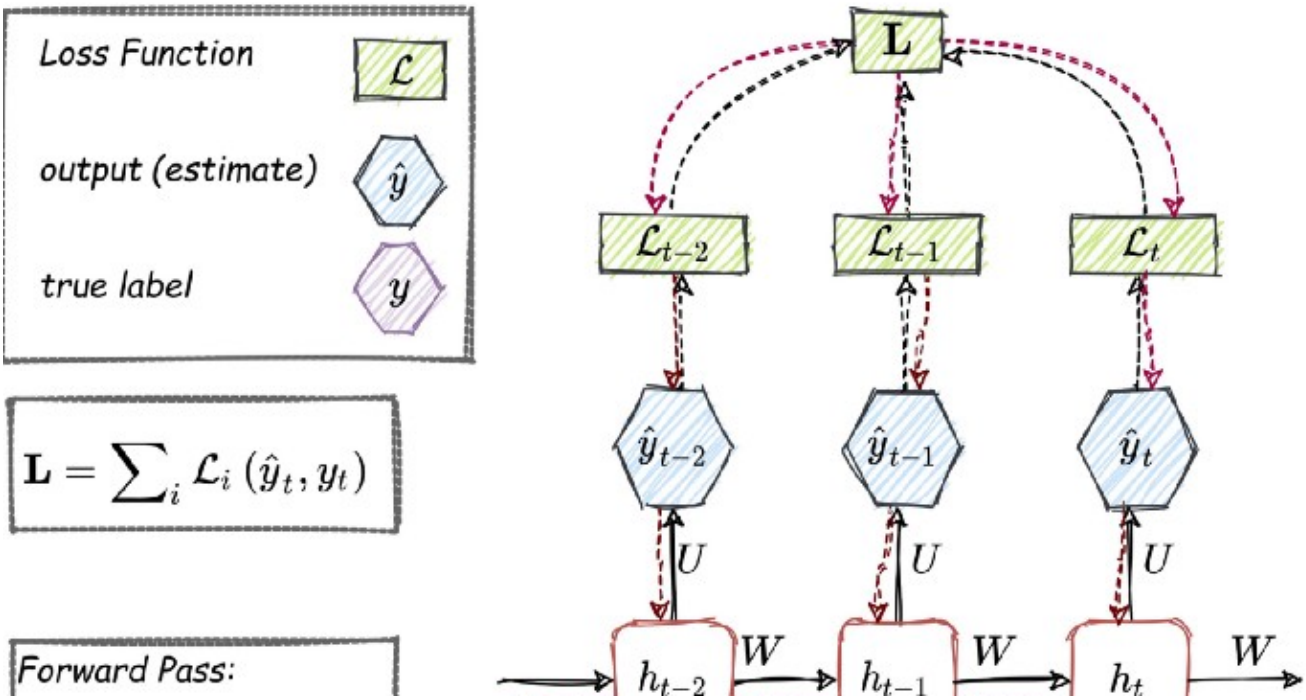
Image by Author

If you are not familiar with RNN, backpropagation, or MLP, please feel free to read references [1]-[3] at the end of the post to fill the gap.

Backpropagation Through Time (BPTT)

Training an RNN is done by defining a loss function (L) that measures the error between the true label and the output, and minimizes it by using forward pass and backward pass. The following simple RNN architecture summarizes the entire backpropagation through time idea.

For a single time step, the following procedure is done: first, the input arrives, then it processes trough a hidden layer/state, and the estimated label is calculated. In this phase, the loss function is computed to evaluate the difference between the true label and the estimated label. The total loss function, **L**, is computed, and by that, the forward pass is finished. The second part is the backward pass, where the various derivatives are calculated.



Get started Open in app

Image by Author

The training of RNN is not trivial, as we backpropagate gradients **through layers** and also **through time**. Hence, in each time step we have to sum up all the previous contributions until the current one, as given in the equation:

$$\frac{\partial \mathbf{L}}{\partial W} = \sum_{i=0}^{T} \frac{\partial \mathcal{L}_i}{\partial W} \propto \sum_{i=0}^{T} \left(\prod_{i=k+1}^{y} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W}$$

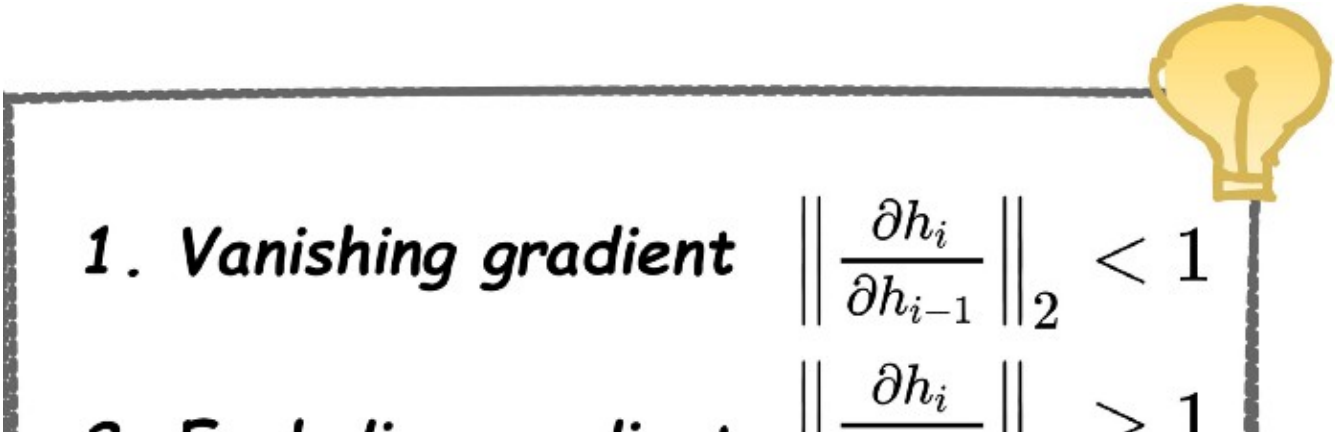
Image by Author

In this equation, the contribution of a state at time step **k** to the gradient of the entire loss function **L,** at time step t=T is calculated. The challenge during the training is in the ratio of the hidden state:

$$\frac{\partial \mathbf{L}}{\partial W} \propto \sum_{i=0}^{T} \left(\prod_{i=k+1}^{y} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W}$$

The Vanishing and Exploding Gradients Problem

Two common problems that occur during the backpropagation of time-series data are the vanishing and exploding gradients. The equation above has two problematic cases:



Get started Open in app

Image by Author

In the first case, the term goes to zero exponentially fast, which makes it difficult to learn some long period dependencies. This problem is called the *vanishing gradient*. In the second case, the term goes to infinity exponentially fast, and their value becomes a NaN due to the unstable process. This problem is called the *exploding gradient*. In the following two sections, we review two approaches to deal with these problems.

Truncated Backpropagation Through Time (Truncated BPTT).

The following "trick" tries to overcome the vanishing gradient problem by considering a moving window through the training process. It is known that in the backpropagation training scheme, there are a forward pass and a backward pass through the **entire sequence** to compute the loss and the gradient. By taking a window, we also improve the training performance from the training duration aspect- where we shortcut it.

This window is called a "*chunk*". During the backpropagation process, we run forward and backward through this chunk of a specific size instead of the entire sequence.

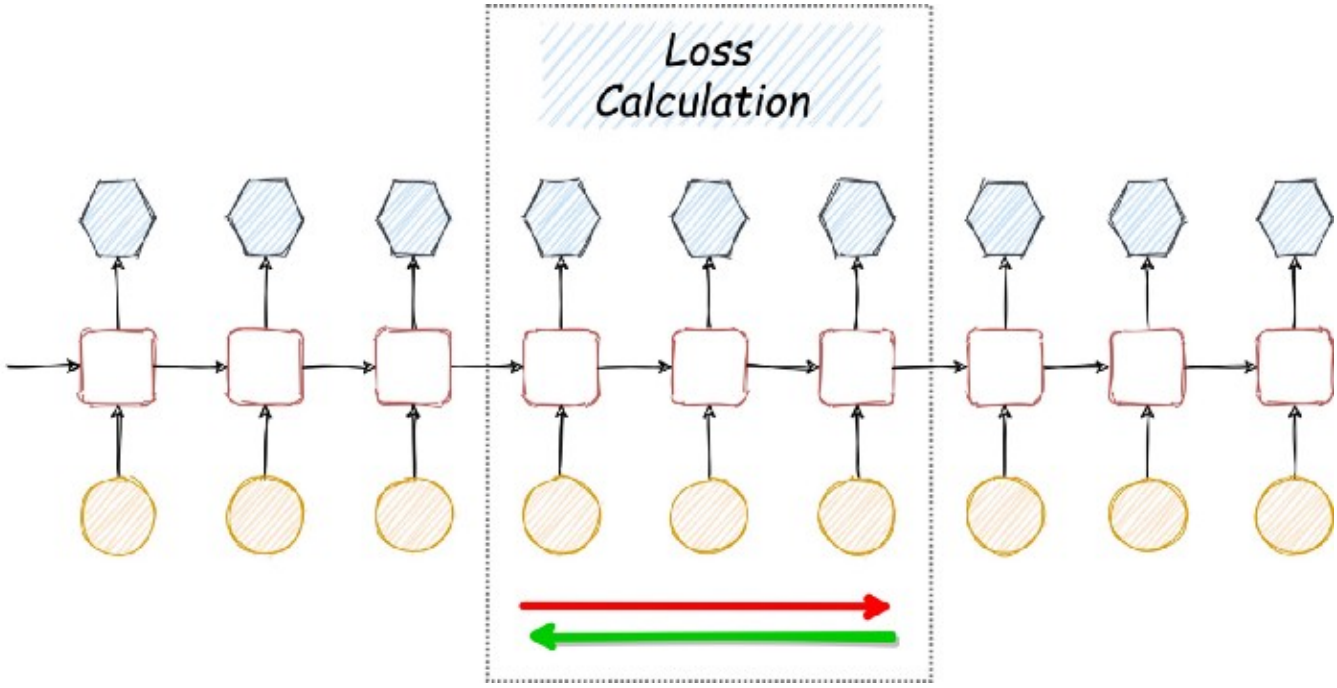


Image by Author

Get started Open in app

minus of this approach is that dependencies of longer than the chunk length, are not taught during the training process. Another disadvantage is the detection of the vanishing gradients. From looking at the learning curve one can assume that the gradient vanishes, but, maybe the task itself is difficult.

For the vanishing gradient problem, many other approaches have been suggested, to mention a few of them:

1. Using ReLU activation function.

2. Long-Short Term Memory (LSTM) architecture, where the forget gate might help.

3. Initialize the weight matrix, **W**, with an orthogonal matrix, and use this through the entire training (multiplications of orthogonal matrices doesn't explode or vanish).

Gradient Clipping

Considering **g** as the gradient of the loss function with respect to all network parameters. Now, define some **threshold** and run the following clip condition in the background of the training process. It is a very simple and very effective condition.

$$clip\left(g, treshold\right) = g \cdot \max\left(1, \frac{treshold}{\|g\|}\right)$$

By applying the gradient clipping, we do not change the gradient direction, but only its magnitude. As the hidden state (**h)** derivative is the part who cause the exploding gradient, it enough to clip the following entity:

$$\frac{\partial h_i}{\partial h_{i-1}}$$

The ***threshold*** is a key parameter the designer should manually define. We aim to choose the highest threshold which solves the exploding gradient problem, by looking at the curve of the gradient norm:

Get started Open in app

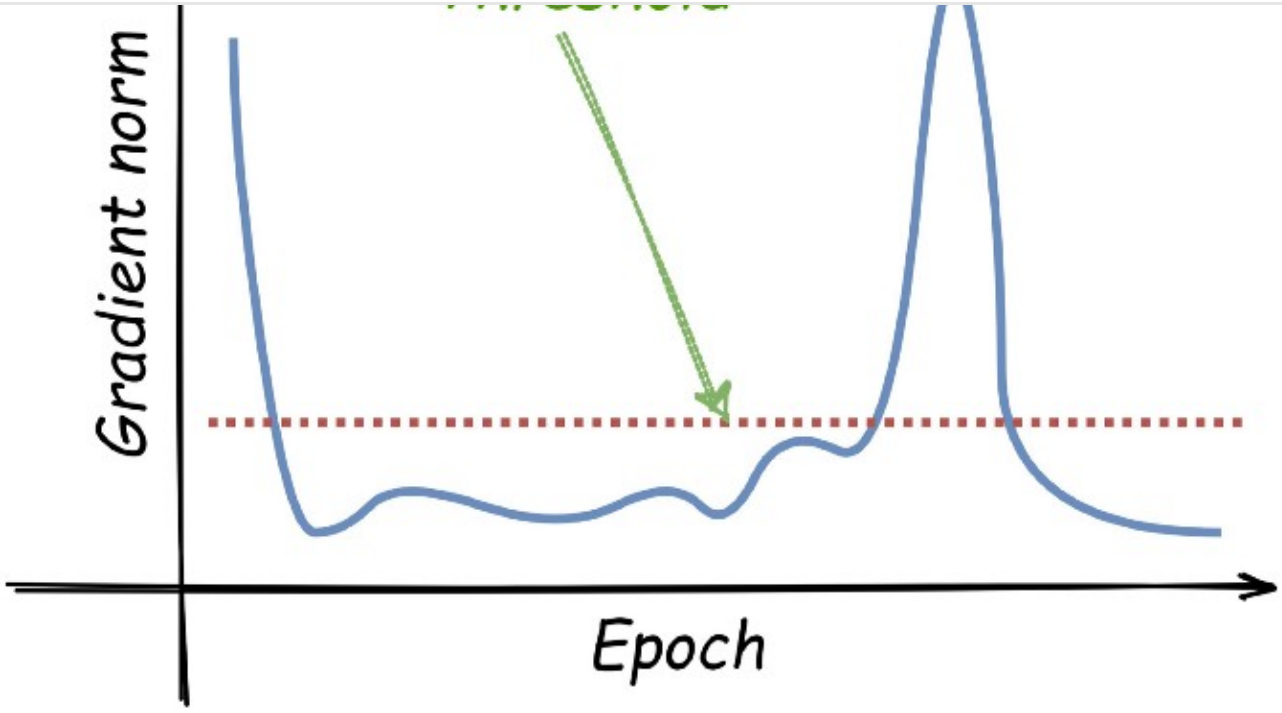


Image by Author

Summary

In this post, we explore the vanishing and exploding gradients problem in simple RNN architecture. These two problems belong to the class of open-problem in machine learning and the research in this pattern is very active. The Truncated BPTT and the gradient clipping approaches were discussed, with some tips for implementation.

About the Author

Barak Or received the B.Sc. (2016), M.Sc. (2018) degrees in aerospace engineering, and also B.A. in economics and management (2016, Cum Laude) from the Technion, Israel Institute of Technology. He was with Qualcomm (2019–2020), where he mainly dealt with Machine Learning and Signal Processing algorithms. Barak currently studies toward his Ph.D. at the University of Haifa. His research interest includes sensor fusion, navigation, deep learning, and estimation theory.

www.barakor.com

https://www.linkedin.com/in/barakor/

References and Further reading

Get started Open in app

[2] Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning. Vibhor Nigam, at Medium. 2018.

[3] Back-Propagation is very simple. Who made it Complicated? Prakash Jay, at Medium. 2017.

[4] Zhang, Jingzhao, et al. "Why gradient clipping accelerates training: A theoretical justification for adaptivity." *arXiv preprint arXiv:1905.11881* (2019).

[5] Chen, Xiangyi, Zhiwei Steven Wu, and Mingyi Hong. "Understanding gradient clipping in private SGD: A geometric perspective." *arXiv preprint arXiv:2006.15429* (2020).

[6] Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." *International conference on machine learning*. 2013.

[7] Ribeiro, António H., et al. "Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness." *International Conference on Artificial Intelligence and Statistics*. 2020.

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. Take a look.

Get this newsletter

Machine Learning Deep Learning Data Science Artificial Intelligence Hands On Tutorials

Get started Open in app

About Write Help Legal

Get the Medium app

Download on the App Store GET IT ON Google Play