Self Normalising Neural Networks

SMAI Team 9 - Project Overview

Introduction and Motivation

Deep Learning has revolutionized vision via convolutional neural networks (CNNs)and natural language processing via recurrent neural networks (RNNs). However,success stories of Deep Learning with standard feed-forward neural networks (FNNs) are rare. FNNs that perform well are typically shallow and, therefore cannot exploit many levels of abstract representations.

We introduce self-normalizing neural networks (SNNs) to enable high-level abstract representations. While batch normalization requires explicit normalization, neuron activations of SNNs automatically converge towards zero mean and unit variance. Self-normalizing neural networks (SNNs) are robust to perturbations and do not have high variance in their training errors . The activation function of SNNs are “scaled exponential linear units” (SELUs), which induce self-normalizing properties. FNNs with batch normalization exhibit high variance due to perturbations. In contrast, SNNs do not suffer from high variance as they are more robust to perturbations and learn faster.

This convergence property of SNNs allows to (1) train deep networks with many layers, (2) employ strong regularization schemes, and (3) to make learning highly robust.

SELU (Scaled Exponential Linear Unit) Activation Function

The activation function is a simple mathematical function that converts a given input into a desired output within a specific range.

SELUs, or Scaled Exponential Linear Units, are activation functions that induce **self-normalization**. SELU network neuronal activations automatically converge to a zero mean and unit variance.

Mathematically, it is expressed as:

$$f(x) = \lambda x \qquad \text{if } x > 0$$

$$f(x) = \lambda \alpha (e^x - 1) \qquad \text{if } x \leq 0$$

SELUs allow to construct a mapping g with properties that lead to SNNs . The activation function is required to have

(1) negative and positive values for controlling the mean

(2) saturation regions (derivatives approaching zero) to dampen the variance if it is too large in the lower layer

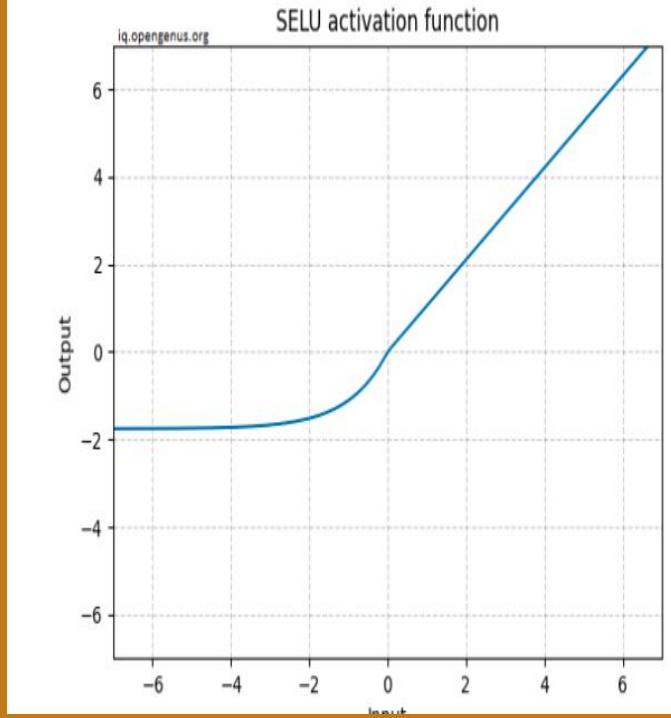
(3) a slope larger than one to increase the variance if it is too small in the lower layer,

(4) a continuous curve. T

Unlike ReLU, it can get below 0, allowing the system to have a zero average output. As a result, the model may converge faster.

SELU is known to be a self-normalizing function, but what is normalization?
Normalization is a data preparation technique that involves changing the values of numeric columns in a dataset to a common scale. This is usually used when the attributes of the dataset have different ranges.

Graphically, it is represented as:



Using Normalization Layers to Improve Deep Learning Models

Normalizing a set of data transforms the set of data to be on a similar scale. For machine learning models, our goal is usually to recenter and rescale our data such that is between 0 and 1 or -1 and 1, depending on the data itself .

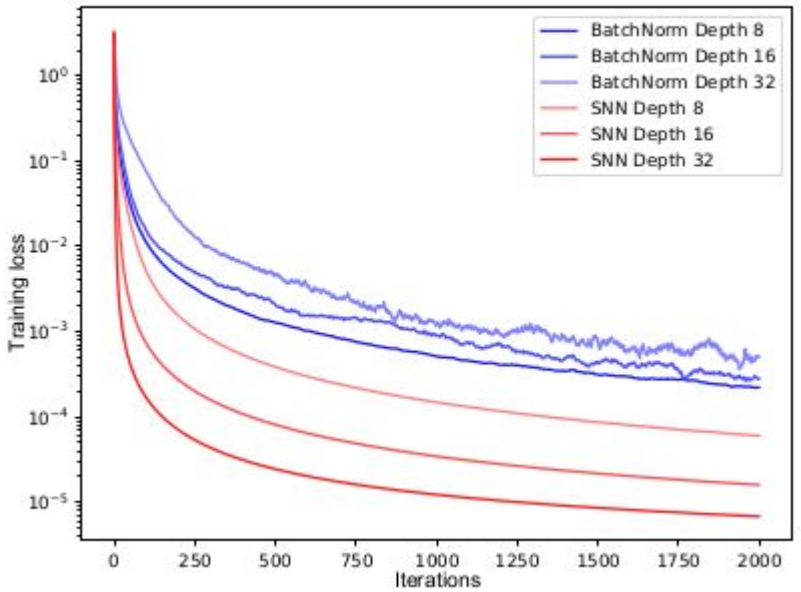
Normalization can help training of our neural networks as the different features are on a similar scale, which helps to stabilize the gradient descent step, allowing us to use larger learning rates or help models converge faster for a given learning rate.

Most of the time , standardising the inputs into each layer help to improve model performance too .

Why SNNs better that FNNs?

FNNs trained with normalization techniques suffer from the perturbations and have high variance in the training error. This high variance hinders learning and slows it down. Furthermore, strong regularization, such as dropout, is not possible as it would further increase the variance which in turn would lead to divergence of the learning process. This sensitivity to perturbations is the reason that FNNs are less successful than RNNs and CNNs.

Self-normalizing neural networks (SNNs) are robust to perturbations and do not have high variance in their training errors . SNNs push neuron activations to zero mean and unit variance thereby leading to the same effect as batch normalization, which enables to robustly learn many layers. SNNs are based on scaled exponential linear units “SELUs” which induce self-normalizing properties like variance stabilization which in turn avoids exploding and vanishing gradients.



FNNs with batch normalization exhibit high variance due to perturbations. In contrast, SNNs do not suffer from high variance as they are more robust to perturbations and learn faster.

Fixed points for Normalised weights

We assume a normalized weight vector w with $\omega = 0$ and $\tau = 1$. For n units with activation xi , $1 <= i <= n$ in the lower layer, we define n times the mean of the weight vector $w \in R^n$ as $\omega := \text{sum}(wi)$ and n times the second moment as $\tau := \text{sum}(wi*wi)$.

We obtain the fixed point equations $\tilde{\mu} = \mu = 0$ and $\tilde{\nu} = \nu = 1$ that we solve for α and λ and obtain the solutions $\alpha_{01} \approx 1.6733$ and $\lambda_{01} \approx 1.0507$, where the subscript 01 indicates that these are the parameters for fixed point (0, 1).

The Jacobian of g has a norm smaller than 1 at the fixed point, g is a contraction mapping and the fixed point is stable , its largest singular value is $0.7877 < 1$. Therefore, (0, 1) is a stable fixed point of the mapping g.

Definition 1 (Self-normalizing neural net). A neural network is self-normalizing if it possesses a mapping $g : \Omega 7\rightarrow \Omega$ for each activation y that maps mean and variance from one layer to the next and has a stable and attracting fixed point depending on (ω, τ) in Ω. Furthermore, the mean and the variance remain in the domain Ω, that is $g(\Omega) \subseteq \Omega$, where $\Omega = \{(\mu, v) \mid \mu \in [\mu \min , \mu \max], v \in [v \min , v \max]\}$. When iteratively applying the mapping g, each point within Ω converges to this fixed point.

Fixed points for Un-Normalised weights

A normalized weight vector w cannot be ensured during learning.

For SELU parameters $\alpha = \alpha_{01}$ and $\lambda = \lambda_{01}$, we show in the next theorem that if (ω, τ) is close to (0, 1), then g still has an attracting and stable fixed point that is close to (0, 1). Thus, in the general case there still exists a stable fixed point which, however, depends on (ω, τ). If we restrict (μ, v, ω, τ) to certain intervals, then we can show that (μ, v) is mapped to the respective intervals.

Variance of neuron activations

Mapping v across layers to a high value corresponds to an exploding gradient . Analogously, mapping v across layers to a low value corresponds to an vanishing gradient. Bounding the mapping of v from above and below would avoid both exploding and vanishing gradients.

From the theorem , : $v\tilde{}(\mu, \omega, v, \tau, \lambda 01, \alpha 01) < v$ and : $v\tilde{}(\mu, \omega, v, \tau, \lambda 01, \alpha 01) > v$, lower bound is 0.24 or 0.16 according to τ and upper bound is 3 .

SELU networks control the variance of the activations and push it into an interval, whereafter the mean and variance move toward the fixed point.

New Dropout

Standard dropout randomly sets an activation x to zero with probability 1 − q for 0 < q <1. In order to preserve the mean, the activations are scaled by 1/q during training.

Dropout fits well to rectified linear units, since zero is in the low variance region and corresponds to the default value. For scaled exponential linear units, the default and low variance value is $\lim_{x \to -\infty} \text{selu}(x) = -\lambda\alpha = \alpha'$. Therefore, we propose “alpha dropout”, that randomly sets inputs to α’.

Conclusion

SNNs do not face vanishing and exploding gradient problems. Therefore, SNNs work well for architectures with many layers, allowed us to introduce a novel regularization scheme, and learn very robustly. On 121 UCI benchmark datasets, SNNs have outperformed other FNNs with and without normalization techniques, such as batch, layer, and weight normalization. SNNs also yielded the best results on drug discovery and astronomy tasks. The best performing SNN architectures are typically very deep in contrast to other FNNs.

SNNs have outperformed state of the art machine learning methods on UCI datasets with more than 1,000 data points.

Remaining work

- Better understanding of paper
- Figuring out implementation details
- Implementation and comparison with other networks

Thankyou