ISPR Midterm 4
Zoneout: Regularizing RNNs by Randomly Preserving Hidden Activations

Diego Arcelli - 647979

University of Pisa

Accademic Year 2022-2023

IN SUPREMÆ DIGNITATIS
1343

UNIVERSITÀ DI PISA

Introduction to the problem

Vanishing gradient in deep neural networks happens when the gradients used to update the network weights become extremely small, making it difficult for the network to learn.

In RNNs this problem affects the ability of propagating information over long sequences, limiting the ability to capture long-term dependencies.

The paper addresses the problem of vanishing gradient in RNNs by proposing a new regularization technique called **zoneout** which aims to favor a better propagation of the gradient during training.

Method description

Similarly to dropout, zoneout injects noise during training, but instead of randomly setting some of the units' activation to 0, it set them to their values at the previous time step.

At each timestep t a binary mask d_t is randomly generated, the neurons for which the mask value is 0 are set to h_{t-1} while the others are normally updated using the transition operator $h_t = \mathcal{T}(h_{t-1}, x_t)$.

$$\text{Dropout:} \quad \mathcal{T} = d_t \odot \tilde{\mathcal{T}} + (1 - d_t) \odot 0$$

$$\text{Zoneout:} \quad \mathcal{T} = d_t \odot \tilde{\mathcal{T}} + (1 - d_t) \odot 1$$

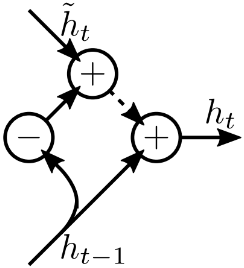
0 is the null operator and 1 is the identity operator.

Method description

Zoneout can be seen as dropout applied in this modified computational graph, in which dropout is applied to $\tilde{h}_t - h_{t-1}$, where $\tilde{h}_t = \mathcal{T}(h_{t-1}, x_t)$.

If we don't apply dropout to $\tilde{h}_t - h_{t-1}$ (dashed connection), then the output is $h_t = (\tilde{h}_t - h_{t-1}) + h_{t-1} = \tilde{h}_t$.

If dropout is applied $\tilde{h}_t - h_{t-1}$ is zeroed and so $h_t = h_{t-1}$.



Key Catch (I)

In LSTM zoneout is applied separately to the memory cell state c_t and the hidden state h_t. First we compute the input, forget and output gates:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \quad f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

Then we compute the LSTM updates:

$$g_t = \tanh(W_x x_t + W_h h_{t-1} + b_h)$$

$$c_t = d_t^c \odot c_{t-1} + (1 - d_t^c) \odot (f_t \odot c_{t-1} + i_t \odot g_t)$$

$$h_t = d_t^h \odot h_{t-1} + (1 - d_t^h) \odot (o_t \odot \tanh(f_t \odot c_{t-1} + i_t \odot g_t))$$

Key Catch (II)

The gates i_t, f_t and o_t and g_t are computed as usual. When we compute c_t according to the mask d_t^c some units are set to their previous value c_{t-1} while for the others we apply the usual LSTM update, controlling with f_t the contribution from the previous state c_{t-1} and with i_t the contribution from the current input g_t.

Also for h_t, according to the binary mask d_h^t, some units are set to h_{t-1} and the others are updated controlling with o_t how much of the cell state c_t we consider to compute h_t.

Note that to compute h_t we don't use the previously computed c_t, but we use $f_t \odot c_{t-1} + i_t \odot g_t$, so that the effect of zoneout on c_t doesn't affect the effect of zoneout on h_t.

Empirical results (I)

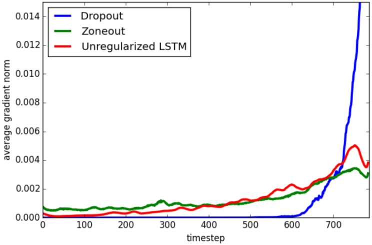
Zoneout is compared to dropout on both character and word level language modeling, and also in image classification on the *p*MNIST dataset, by training different RNNs using both regularization methods.

On every task the results show that every network when trained using zoneout achieves better test performances than when trained with dropout.

They also compare their results with the performances on the same datasets of some SoTA recurrent architectures (like hierarchical multi-scale RNN). The networks trained with zoneout are not as good as all of those SoTA models, but they achieve close results.

Empirical results (II)

The plot shows the average gradient magnitude at time t of the loss w.r.t. c_t after one epoch of training on the pMNIST dataset. We can see that for the model using zoneout in the early timesteps the gradient is higher, hence it has a better gradient propagation to the early timesteps.



Comments

The experimental results provided in the paper show that using zoneout as regularizer leads to slightly better performance than using dropout, while having a better gradient propagation which helps in mitigating the vanishing gradient problem.

Zoneout is a very simple technique and easy to implement. One limitation is that unlike other regularization techniques (like dropout) is designed specifically for RNNs and it cannot be applied to different kind of neural networks.

Bibliography

[1] David Krueger, Tegan Maharaj, János Kramár, Mohammad Pezeshki, Nicolas Ballas, Nan Rosemary Ke, Anirudh Goyal, Yoshua Bengio, Aaron Courville, and Chris Pal.
Zoneout: Regularizing rnns by randomly preserving hidden activations.
arXiv preprint arXiv:1606.01305, 2016.