**PReLU**

FSRCNN

PReLU: For the activation function after each convolution layer, we suggest the use of the Parametric Rectiﬁed Linear Unit (PReLU) [1] instead of the commonly-used Rectiﬁed Linear Unit (ReLU). They are different on the coefﬁcient of the negative part. For ReLU and PReLU, we can deﬁne a general activation function as f(xi) = max(xi,0) + aimin(0,xi), where xi is the input signal of the activation f on the i-th channel, and ai is the coefﬁcient of the negative part. The parameter ai is ﬁxed to be zero for ReLU, but is learnable for PReLU. We choose PReLU mainly to avoid the “dead features” [2] caused by zero gradients in ReLU. Then we can make full use of all parameters to test the maximum capacity of different network designs. Experiments show that the performance of the PReLU-activated networks is more stable, and can be seen as the up-bound of that for the ReLU-activated networks.

//////

As suggested by He et al. [1], we used PReLU activation in the generator and discriminator for all layers, except for the output which uses a Tanh function, since this strategy allowed the model to learn quickly [3]

/////

All PReLUs were initially set to 0.33. The stride of transposed convolution was 2 to ensure that each transposed convolution would magnify the image twice. The kernel size of each convolution was 3 x3. We used the same strategy as He *et al.* [1]for convolution weight initialization.

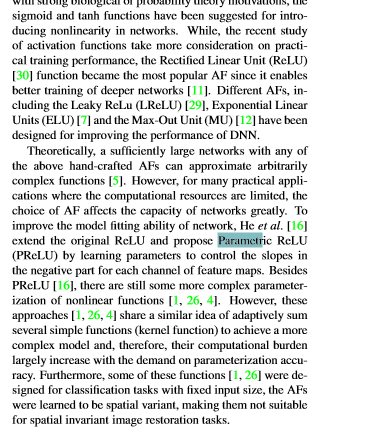
//////

The combination of these models can massively improve the SR performance of the proposed model. It is worth to mention that we used same activation function for both the networks as Parametric rectified linear units (PReLU).

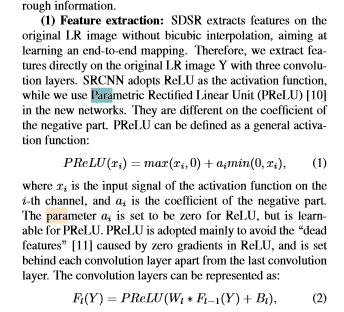
/////////

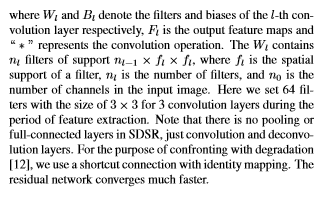
In DSIRSR model, the activation function of each layer adopts Parametric Rectified Linear Unit (PReLU) [], which can be regarded as ReLU activation function with correction parameters. Compared to the ReLU activation function, the PReLU activation function only adds a small amount of computation to achieve a higher accuracy and can avoid the “dyingReLU” phenomenon caused by ReLU.

//////////////

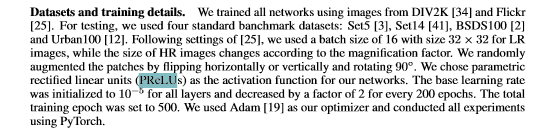


//////

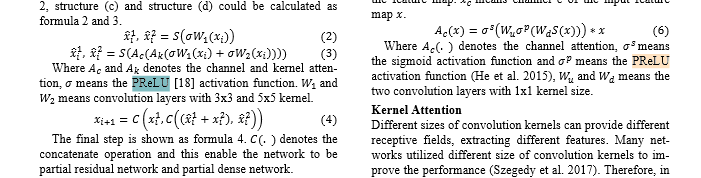




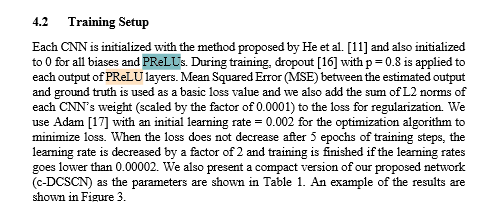
///

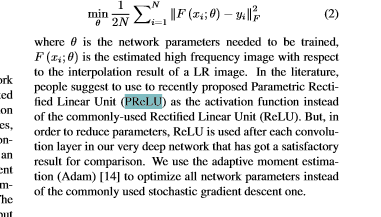


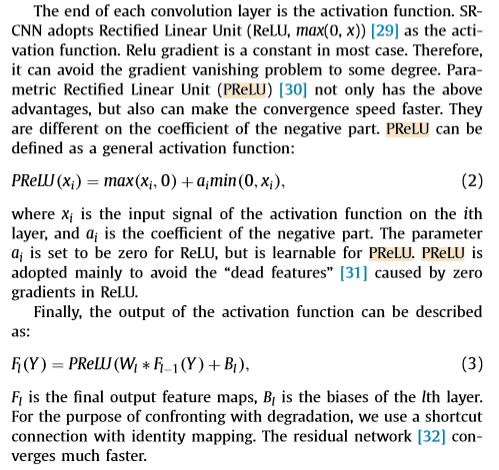
//////////

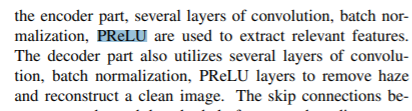


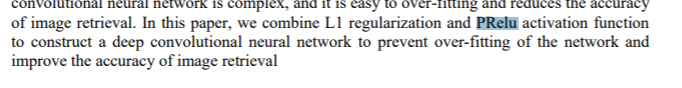
////

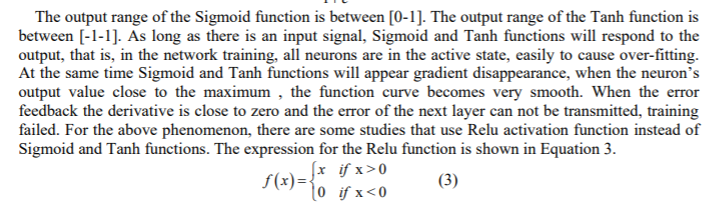


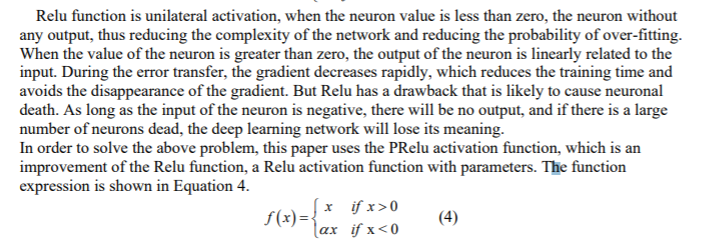


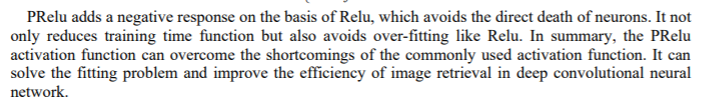




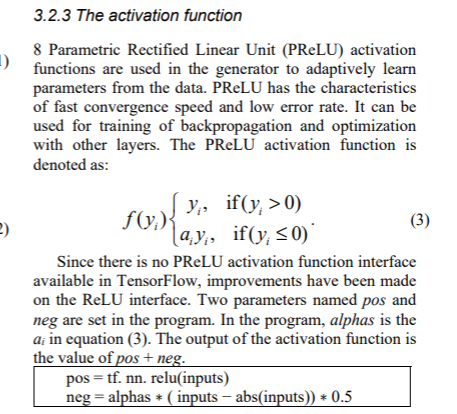








Furthermore, we use the Parametric Linear Unit (PReLU) to instead of ReLU as the activation function of our convolution layer. Since PReLU has a learnable coefficient for the negative part of features, it can avoid the “dead feature” caused by zero gradients in ReLU. Accordingly, we can make full use of all parameters to obtain the maximum capacity of our networks.



References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026-1034.

[2] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014, pp. 818-833: Springer.

[3] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434,* 2015.