Studying the Effect of Skip Connection

Sihao Ying
Shanghai Jiao Tong University
yingsihao@sjtu.edu.cn

Yiran Wu
Shanghai Jiao Tong University
yiranwu@sjtu.edu.cn

Abstract

In this report, we study the effect of skip connection in neural networks. We try skip connections on fully-connected and convolutional networks. We evaluate the performance on EMNIST hand-written letters dataset [1] and CIFAR-10 [4] dataset respectively. Our experiments show that adding skip connections doesn't improve the performance of multi-layer perceptron significantly on both dataset. But for convolutional neural network, this technique helps improve the accuracy considerably by trading off accuracy and training with feature reuse.

1. Introduction

With the progress of research on deep learning, vanilla network architecture was proved inferior in many cases, encouraging architectural attempts to better structure neural networks. Some outstanding attempts, such as ResNet [2] or recently DenseNet [3], have drawn people's attention to skip connections. The skip connections are links between non-adjacent layers. Experimental results shows that they are simple yet effective and is applicable under various circumstances. There are three major motivation of introducing skip connections:

- Creating gradient shortcut.

  First proposed in ResNet, this kind of links let inputs of layers flow directly to outputs. They let gradients pass without minification in back propagation, which eases gradient vanishing problem and thus enables training of very deep networks. It's clear why this works and experimenting requires to train deep networks which is time-consuming. So we decided not to examine this.

- Complementing lost information.

  Skip connections are also used in tasks like auto-encoder or segmentation where the neural network is asked to predict a map with its locations aligned to locations in input. Typical network architectures in this case looks like an hourglass. The size of feature map is first shrunk and then upsampled to its original scale. And the shrinking part and upsampling part are usually symmetric. People came up with the idea of adding skip connection between corresponding layers in two parts to improve the final result. Here's our intuition of why this might work: During shrinking stage the feature map is compressed by max pooling, which keeps the maximum activation but throws away the information of the position it comes from. And in upsample stage we have to recover the position information. Going back to where we throw them away might give some clue about what the position should be. That leads to the idea of linking layers in two stages.

- Concatenating shallow layer features with deep ones.

  Concatenating feature maps of shallow layers with feature maps of deep layers is another way of introducing skip connections, as DenseNet proposed. **And it's what we are interested in for this work**. In convolutional neural networks, features in shallow layers typically correspond to microscopic patterns such as colors or edges. Deeper layers tend to learn more about semantics, objects, and image layout. DenseNet try combining information from different perspective to help classification. We are curious about the effect of this combination. Does it work the same way as in segmentation? Will it also work on fully connected nets? Why is feature of shallow layers also applicable to deep layers? Why not add more filters and let the network learn automatically to reproduce former features if needed? We plan to do experiment on this kind of skip connection and examine how it works.

2. Experiment

2.1. Multi-layer Perceptron for Hand-written Letters Classification

First, we try to study whether skip connections have positive effect on multi-layer perceptron in the task of hand-written letters classification. We construct the network architecture illustrated in Figure 1 (just ignore the red part now, they are for skip version). The dataset we use is EMNIST containing 60,000 28 by 28 grayscale images of handwritten digits.

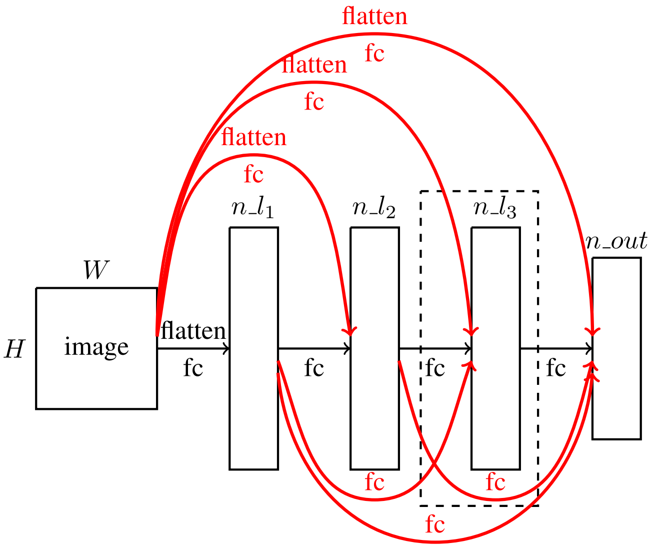

Figure 1. Multi-layer Perceptron for EMNIST

The fully connected layer in dashed box is optional, that is, omitting this layer we will get a 3-layer network while keep it we will have a 4-layer one. After each layer, we apply ReLU as activation function. n_l1, n_l2, n_l3, n_out means the number of neurons in layer 1, 2, 3 and the output layer. In our experiment, we set $n\_l1 = 128$, $n\_l2 = 256$, $n\_l3 = 256$, $n\_out = 26$.

Next, we insert skip connections in this vanilla network. For each pair of non-adjacent layers i and j, we will insert fully connections between layer i and layer j, which means every neuron in layer j will be connected to every neuron in layer i. We illustrate this in Figure 1 with the red part.

Then we compare the performance of vanilla network and skip-connected network. For each network, we train it on training set from the very beginning and tune the regularizer to get a highest accuracy on validation set. We use mini-batch stochastic gradient descent as our optimization method. The results are listed as follows:

	train(3-layer)	val(3-l)	train(4-l)	val(4-l)
skip	91.0	89.0	91.6	89.1
vanilla	90.4	88.6	92.0	90.0

Table 1. Results of two MLPs on EMNIST

As we can see, for the 3-layer network, the skip version performs slightly better than the vanilla version. However, for the 4-layer one, the vanilla one outperforms the skip one, which is quite surprising. In our expectation, with higher capability, the skip-connected network should always perform better. We first doubt that the weights in skip connections might be too small as if they weren't there. But the magnitude of weights in skip connections turn out to be within reasonable range. Another suspect is that we fall into local minimum. So we design a different training strategy for skip-connected network. First, we disable all the skip connections and train the network as a vanilla one. After it converges, we keep all the weights, enable the skip connections and train the whole network to convergence. This brings us the following result:

	train(3-layer)	val(3-l)	train(4-l)	val(4-l)
skip*	91.2	89.4	92.5	90.9

Table 2. Results of Skip Version Using New Training Strategy

which is better than the vanilla one with both 3 and 4-layer architectures as we expected, but not significantly.

Analysis. Introducing skip connection does not give much performance gain in this setting. It might even degrade the validation accuracy. This is strange because the skip network can choose to ignore all skip links and only use links that exist in vanilla networks. And we have some suspicion why skip connection does not work:

- Too simple dataset.
  The 28 by 28 grayscale image might be so simple that features of vanilla network are good enough to perform classification. No need for additional features.

- Too simple networks.
  To gain rich information by combining features, the information sources should have enough variety. It might be that our network is too simple. The layer before softmax is still detecting local patterns, much like the first layer. And combining these homogenous information is not helpful.

- Not the right task.
  We are wondering whether this kind of feature concatenation makes sense in classification problems. It is proved to be helpful in segmentation task, serving as completion of lost information. But in classification task people does not care the lost information like positions. We only care about does there exist one, no matter where. So the additional information does not make difference.

- Not applicable to fully connected networks
  Literatures about skip connection are usually based on convnets. It might not work for fully connected ones. One possible reason for that is the inherent difference of the two architectures. Neurons in convnets have spatial locality. Every neuron has its own active region on original image. And feature maps from different layers have such inherent alignment where neurons at same location in the feature map correspond to same

region on the image. Such relation between features of different layers probably can make the feature integration more productive. While in fully connected networks, every neuron are linked to all neurons in the previous layer. There's no clear relation between activations of two layers. The neural network might have difficulty in figuring out how to integrate them.

2.2. Multi-layer Perceptron on CIFAR-10

To verify our first doubt, we make the data more complex by using CIFAR-10. Our network architecture is the same as the one in Figure 1. We set $n\_l1 = 512$, $n\_l2 = 256$, $n\_l3 = 256$, $n\_out = 10$. After training two version of networks, we get the following results (the skip* means the skip version using the special training strategy we designed while the skip means training from the very beginning):

	train(3-layer)	val(3-l)	train(4-l)	val(4-l)
skip	64.9	64.1	67.8	66.6
vanilla	64.0	63.2	66.5	65.8
skip*	64.9	64.2	68.0	66.7

Table 3. Results of two MLPs on CIFAR-10

From the table, we can see skip connections can improve the accuracy of MLP on CIFAR-10 dataset, however, still not significantly.

2.3. Convolutional Neural Network on CIFAR-10

From the above experiments, we suspect that skip connects are not so applicable to fully connected networks. So we decide to add them on CNN and see how it effects the classification accuracy. The CNN architecture we use construct is defined in Figure 2 (just ignore the red part now, they are for skip version).

As we discussed before, it's reasonable to integrate features from different layers in CNN, so we design a special rule to add skip connections. As the red part illustrates in Figure 2, we add skip connections from layer 1 to layer 4, layer 2 to layer 5, layer 4 to layer 7, layer 5 to layer 8, respectively. Take the skip connection from layer 1 to 4 for example. On this connection, tensor in layer 1 with shape [32, 32, 32] will firstly go through a 2 by 2 maxpooling operation and become shape of [16, 16, 32]. Then we concatenate it to the original tensor in layer 4 whose shape is [16, 16, 64] and get a tensor with shape [16, 16, 96], which will be the new tensor in layer 4.

We train two versions of networks and test their accuracy on validation set. The results are listed in Table 4.

	train	val
skip	84.2	83.5
vanilla	81.7	80.9

Table 4. Results of two CNNs on CIFAR-10

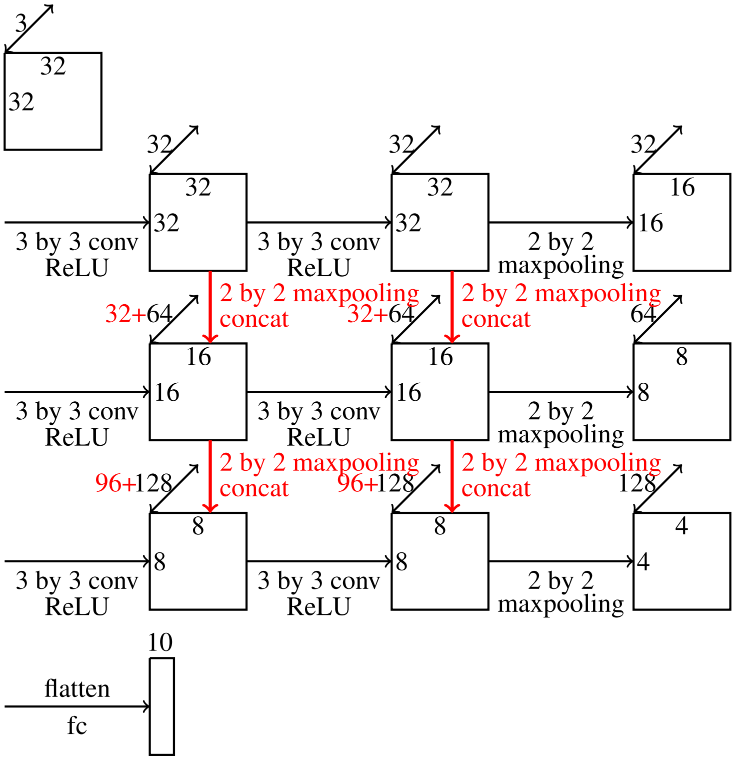

Figure 2. Convolutional Neural Network for CIFAR-10

The table shows the performance of our CNN can be improved considerably with skip connections. It's a good news, but we still have one thing to do. In skip-connected version, some layers (layer 4, 5, 7, 8) have more channels than in vanilla version, which means higher capability and seems unfair. So for the vanilla CNN, we increase the number of feature extractors in layer 4 and 5 to 96, and layer 7, 8 to 224. After training this CNN, we get the following result:

	train	val
vanilla*	84.9	84.1

Table 5. Results of Reinforced CNN on CIFAR-10

As we can see, the modified vanilla CNN performs slightly better than our skip version. But it's within our expectation. In skip-connected CNN, the additional channels are fixed by us while in reinforced vanilla CNN, they can be learned and calculated by the network itself. Obviously the latter is much more flexible which makes use of the additional feature extractors to detect and learn something important for its classification task.

Analysis. However, one thing to point out is the training time of reinforced vanilla CNN is nearly 2 times of that of skip-connected CNN because due to parameters in additional filters. But the accuracy is just slightly better.

3. Conclusion

In experiments, our finding is that adding skip connections which concatenate feature of multiple layers in CNN will help improve classification performance. Adding same amount of learnable filters will give slightly better performance than skip connection at the cost of several times longer training time. In other words, reusing feature from former layers makes sense, but learnable features fit in the model better with longer training time. It's a tradeoff between training time and accuracy.

So in our opinion, the essence of this kind of skip connection is not the observation that replicating features will help. It's the idea of balancing accuracy and training time by feature reuse that really matters.

References

[1] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[4] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research).