Resnet

Motivation

- For neural networks, is it the deeper the better?

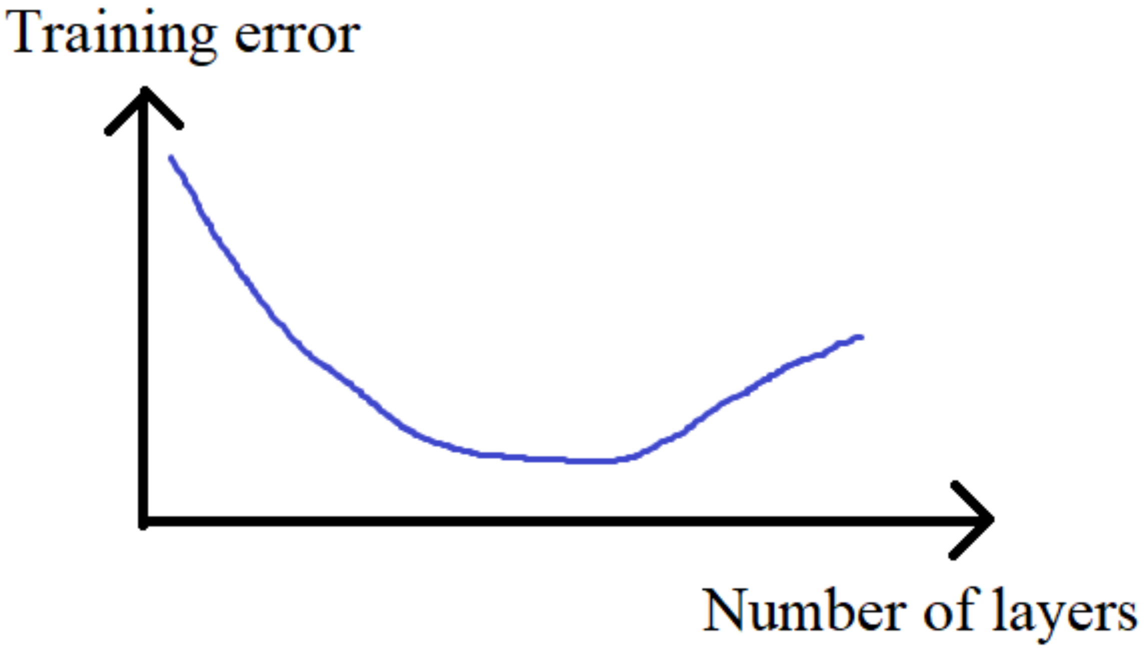
Motivation

- For neural networks, is it the deeper the better?

- Not really.

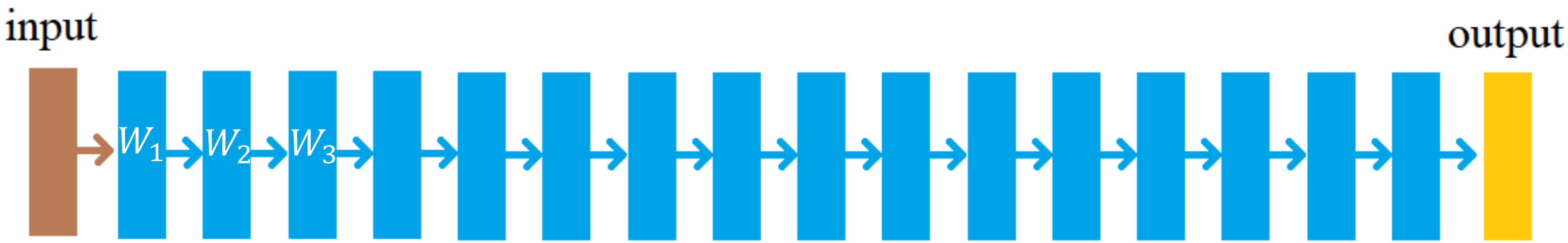
  - It is counterintuitive, but the training error actually increases when the network is too deep.

  - It is not over-fitting. The training error increases not the testing error.



Vanishing & exploding gradients

- Consider a very deep neural network
    - In fact, this network is not deep at all. Nowadays networks in CV normally contain 100+ layers.
  - We do not use activation layers for simplification.
  - The output will be $Y = W_l W_{l-1} \ldots W_2 W_1 \ X$



Vanishing & exploding gradients

- Consider a very deep neural network
  - $Y = W_l W_{l-1} \dots W_2 W_1 \; X$

  - Imagine what if we initialize W with matrix αI
  - $Y = \alpha^l I^l \, X = \alpha^l X$

  - If we initialize W with matrix $0.5I$
  - $Y = 0.5^l \, X,$ if $l = 50, Y = 0.00000000000000888 \, X$ (vanishing)

  - If we initialize W with matrix $1.5I$
  - $Y = 1.5^l \, X,$ if $l = 50, Y = 637621500.214 \, X$ (exploding)

Vanishing & exploding gradients

- Each W is a little small $\rightarrow$ The output is very small
- Each W is a little big $\rightarrow$ The output is very big

- The output will increase/decrease exponentially.
- The derivatives (gradients) will also increase/decrease exponentially.

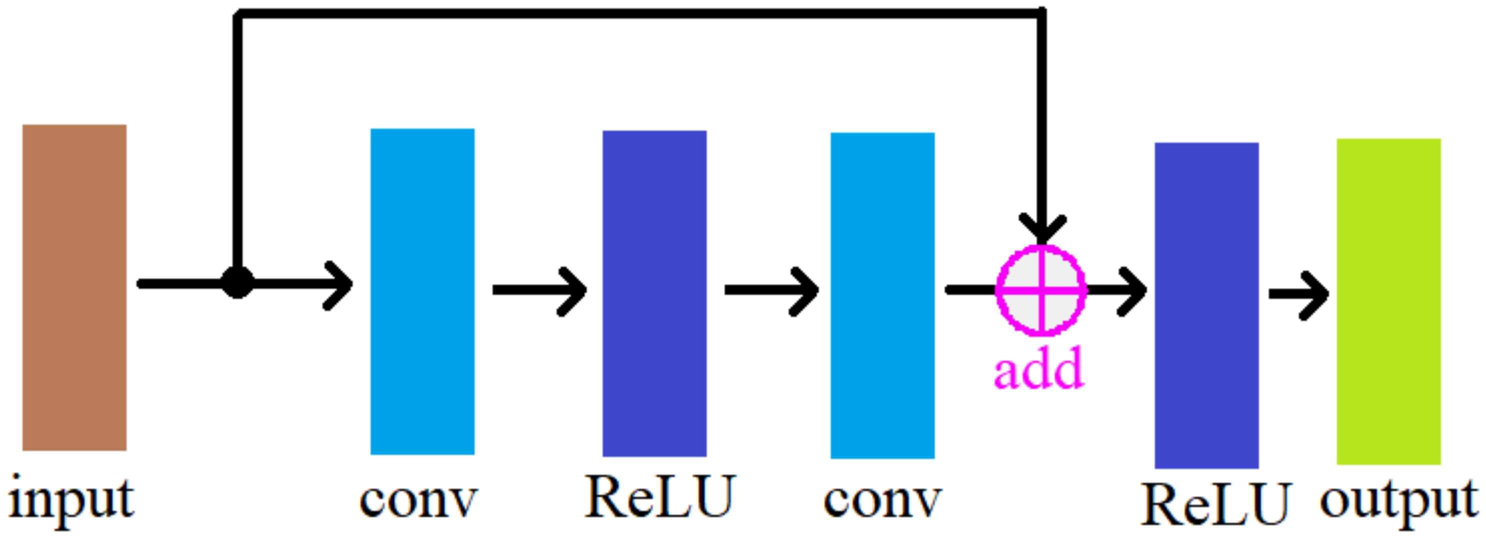
Vanishing & exploding gradients

- First, we need careful initialization of the weights before training.
  - There are many different kinds of initializers
  - Try them in your assignments

- This does not prevent the network from killing itself during training.
  - Batch normalization
  - Leaky ReLU
  - Resnet (Residual Network)

Residual blocks

- Key: shortcut (or skip connection)
- $Y = relu(X + f(X))$



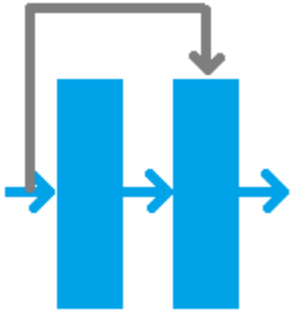
Residual blocks

- $Y = relu(f(X))$ "plain block" without shortcut
- $Y = relu(X + f(X))$ Residual block

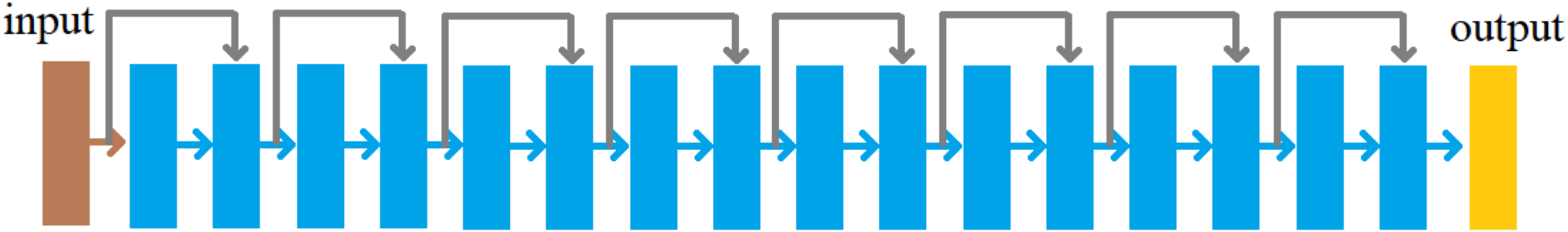
- In the worst case, the layer might want an identity transformation, so that the network is equivalent to a shallower version.
  - For "plain block" there is a construction such that $f(X) = X$
  - But it is hard for the optimizer to make it happen
  - It is easier for Residual block, it can simply set $f(X) = 0$

Residual Network (Resnet)

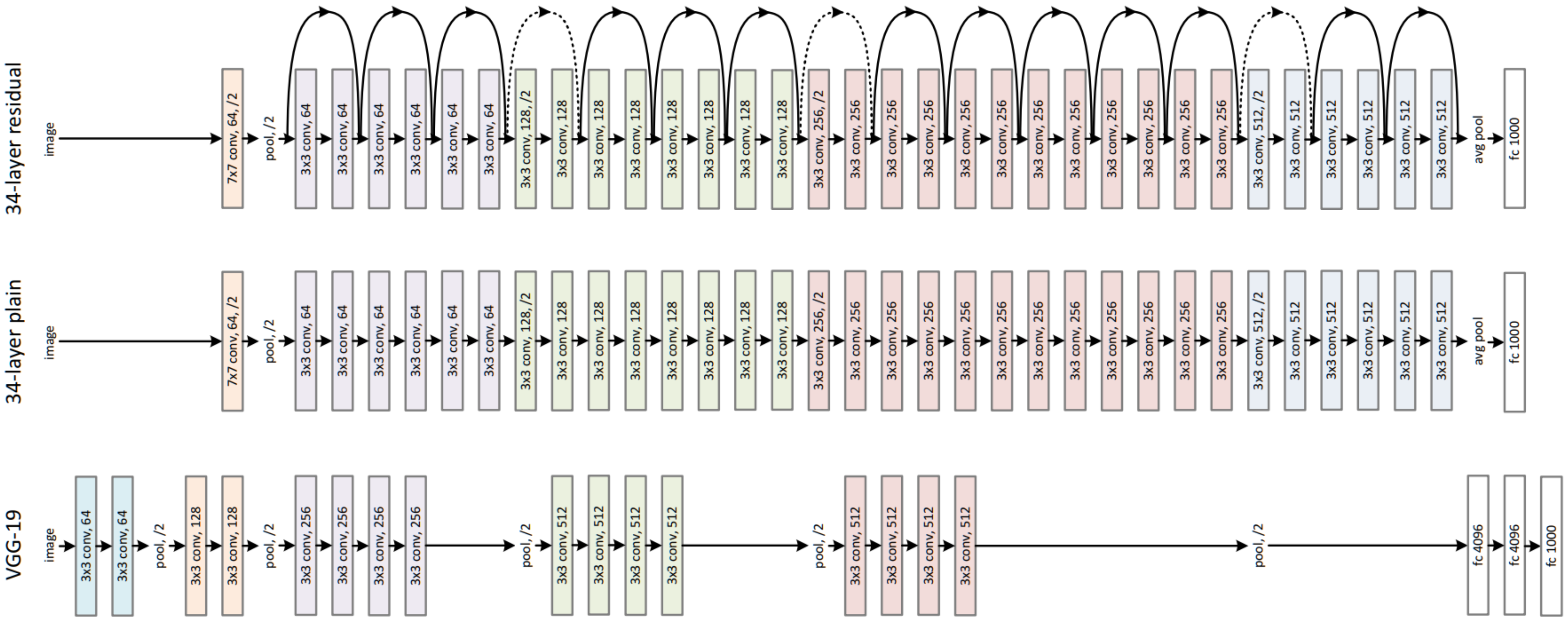
- Stacking Residual blocks together



A Resnet block
(Arrow points to the middle of the second layer because ReLU is done after addition)



Residual Network (Resnet)



Residual Network (Resnet)

- Allow training of very deep neural networks (1000+ layers)

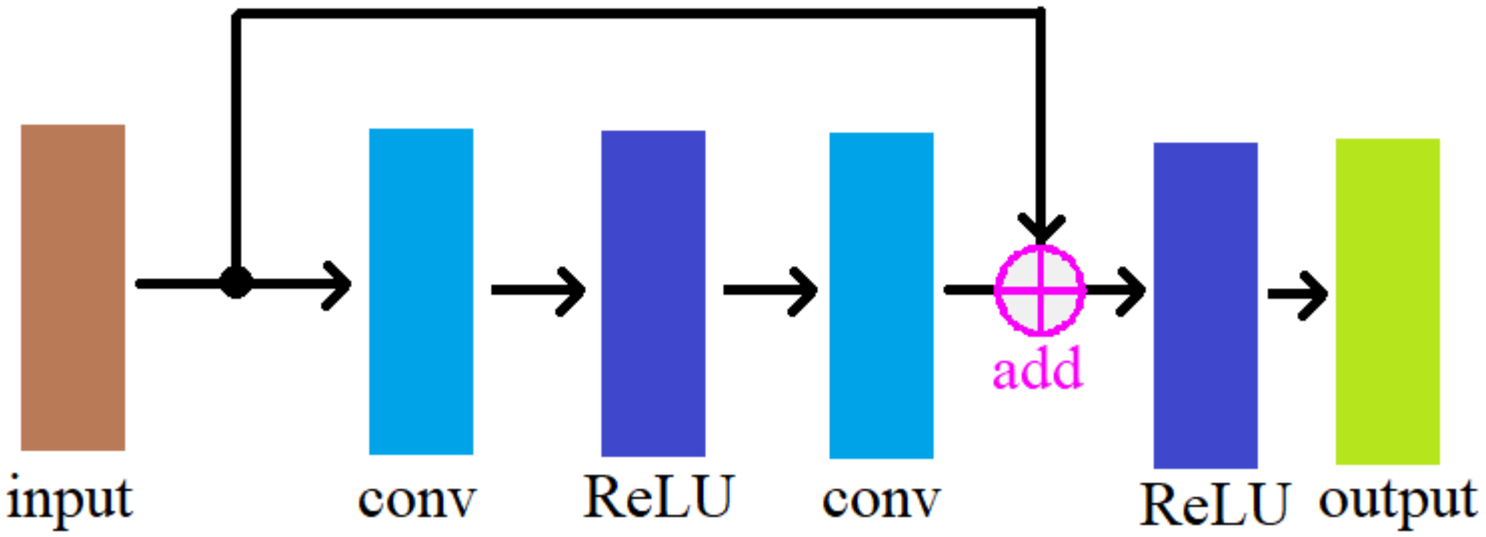
- The performance is no worse than the shallower versions of itself.

  - Identity function is easy for residual blocks to learn
  - $Y = relu(X + f(X))$

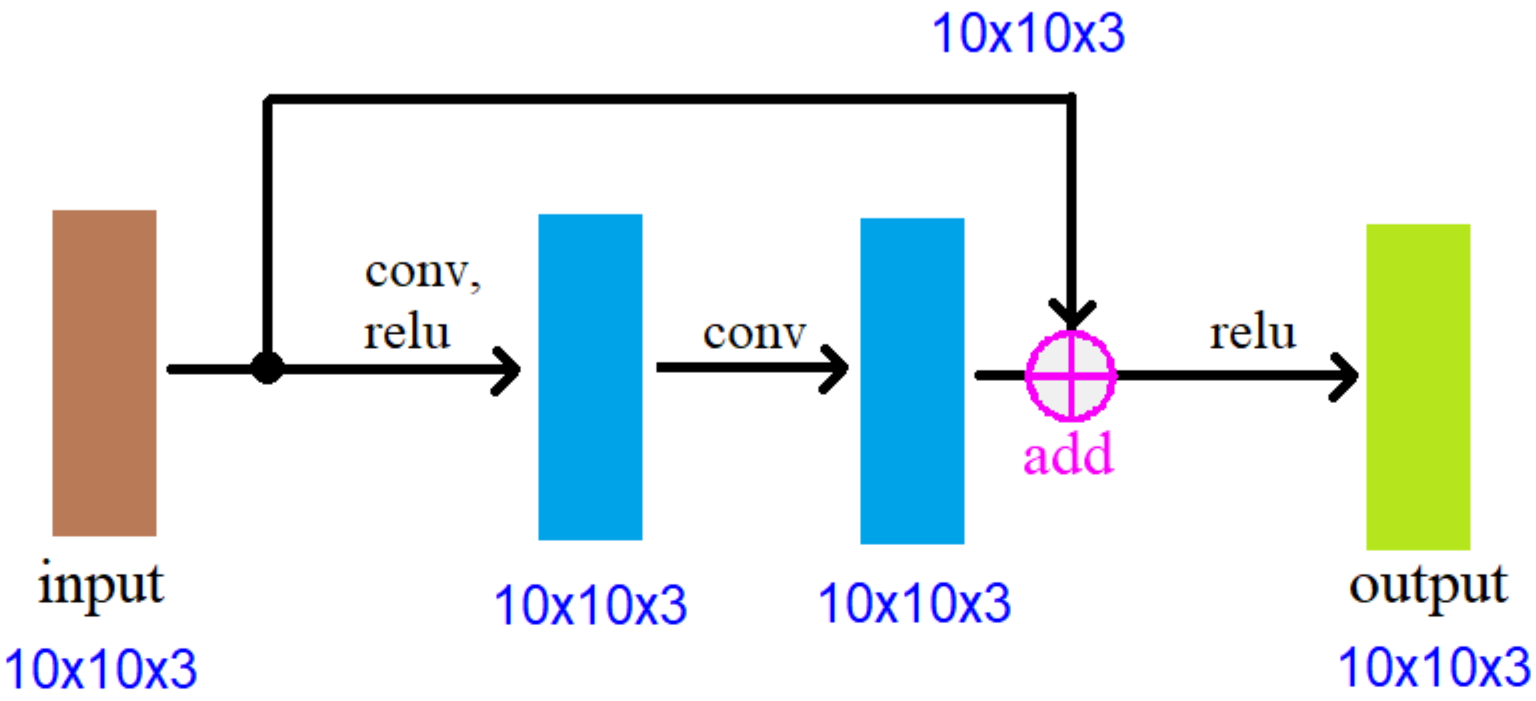
  - Adding more layers won't hurt the performance

Residual Network (Resnet)

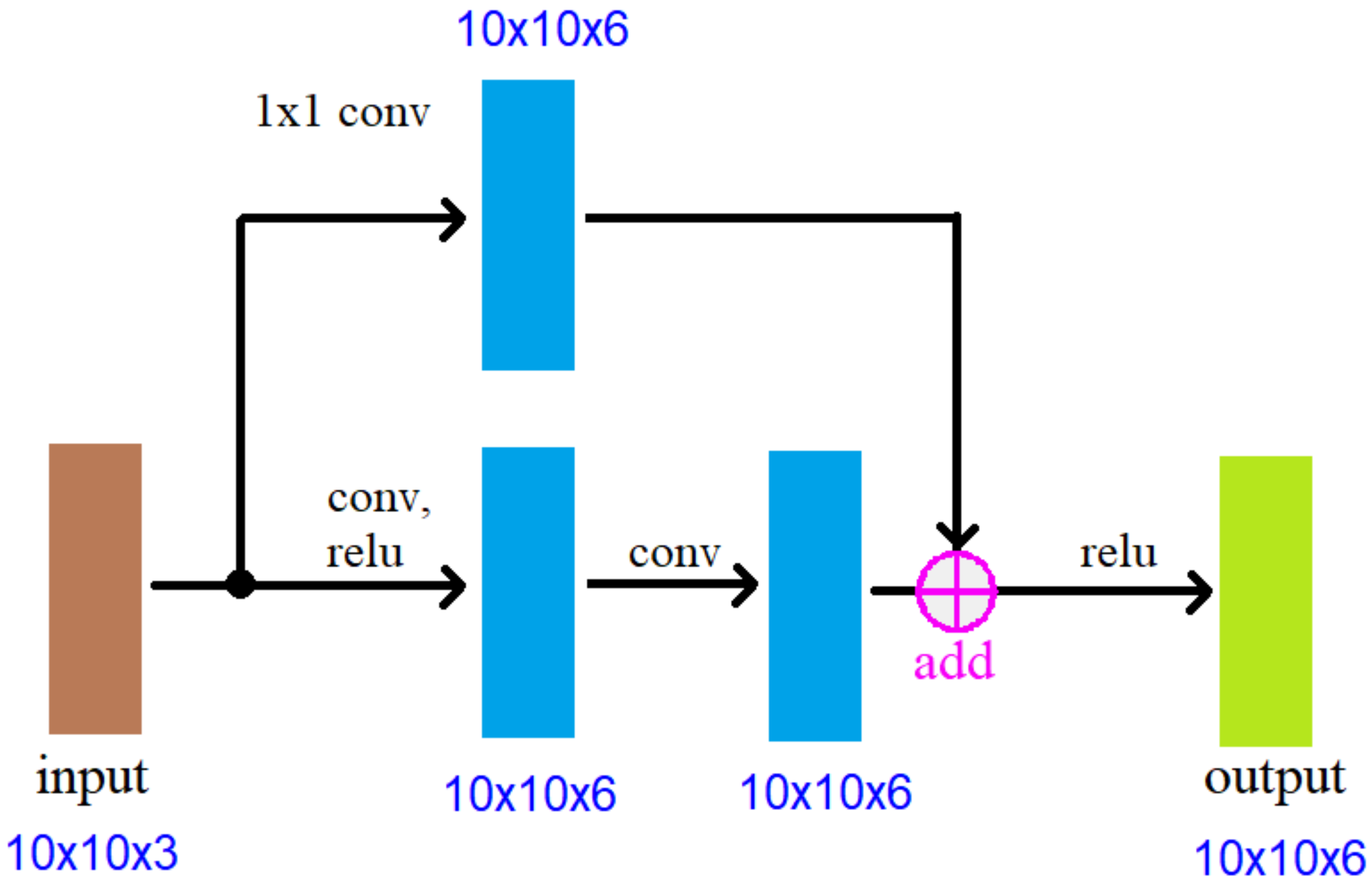
- We assume the input and output have the same dimensions.
- What if their dimensions are different?
- In CNN specifically
  - What if the channel numbers don’t match?
  - What if the image sizes don’t match?



Standard residual block



Change channel number



Change channel number + downsampling