



CS 329P : Practical Machine Learning (2021 Fall)

Residual Connections

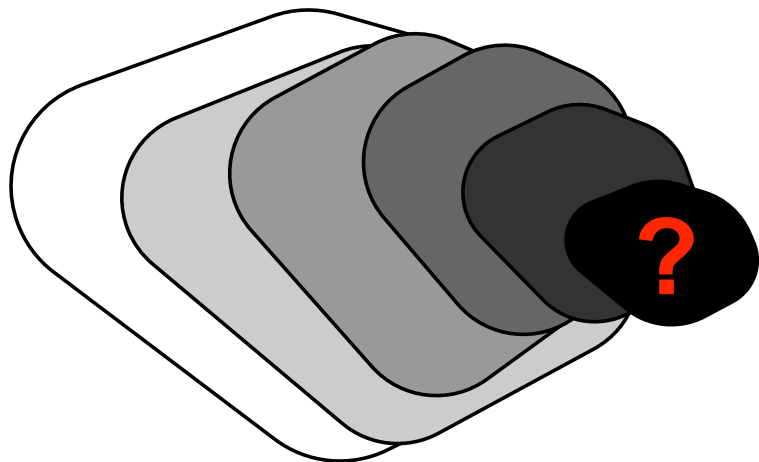
Qingqing Huang, Mu Li, Alex Smola

<https://c.d2l.ai/stanford-cs329p>

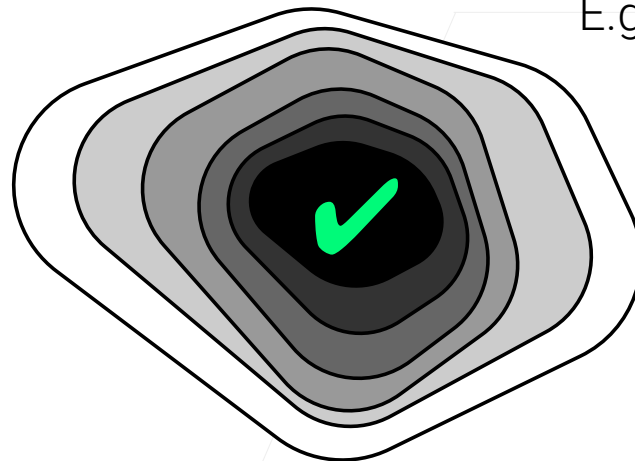
Does Adding Layers Improve Accuracy?



- Add more layers may decrease accuracy



generic function classes



E.g. boosting

nested function classes



Residual Connections

- Adding a layer f changes the function class to $f(g(x))$
- We want to add to the function class $f(g(x)) + g(x)$
 - Larger gradient, easier to train

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g(x)} \frac{\partial g(x)}{\partial x}$$

Matmul may cause small gradients

$$\frac{\partial (f(g(x)) + g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial x} + \frac{\partial g(x)}{\partial x}$$

Gradients for a shallower network

- Boosting over features vs boosting over labels discussed before

ResNet - de facto CNN Architecture



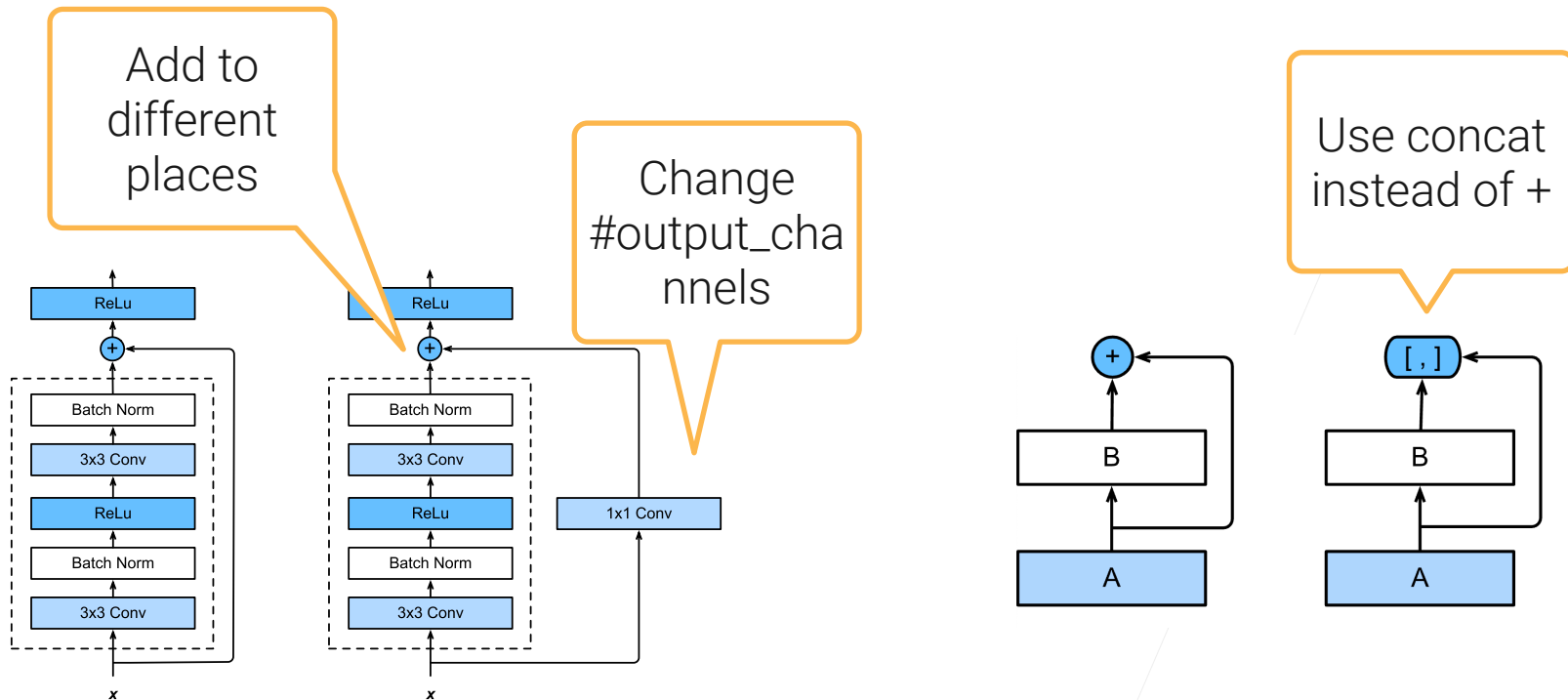
- CNN with stacked Residual blocks

```
class ResidualBlock(nn.Module):
    def __init__(self, input_channels, num_channels):
        super().__init__()
        self.conv1 = nn.Conv2d(
            input_channels, num_channels, kernel_size=3, padding=1)
        self.conv2 = nn.Conv2d(
            num_channels, num_channels, kernel_size=3, padding=1)
        self.bn1 = nn.BatchNorm2d(num_channels)
        self.bn2 = nn.BatchNorm2d(num_channels)

    def forward(self, X):
        Y = F.relu(self.bn1(self.conv1(X)))
        Y = self.bn2(self.conv2(Y))
        return F.relu(Y + X)
```

Full code: http://d2l.ai/chapter_convolutional-modern/resnet.html

Variants



Summary



- Residual connection allows to pass internal layers
 - Make very deep NN possible (people tried CNNs with 1K layers)
 - Build nested function classes
 - Boosting over the features