# ECE472 – Quiz 4

Jonathan Lam

October 1, 2020

1. *Explain how the main technique from "Delving Deep into Rectifiers" may alleviate a core issue in the training scheme for "Going Deeper with Convolutions." (Think about what they did to "get-it-to-work")* (arXiv:1502.01852, arXiv:1409.4842)

   The authors of the Inception network ("Going Deeper with Convolutions") cite that "Given the relatively large depth of the network, the ability to propagate gradients back through all the layers in an effective manner was a concern." This is not only a problem with the Inception, but a general concern with deeper networks in general. To combat this, the authors of the Inception network "add[ed] auxiliary classifiers connected to these intermediate layers, we would expect to encourage discrimination in the lower stages in the classifier, increase the gradient signal that gets propagated back, and provide additional regularization." (Another way to mitigate this is by using skip connections like a ResNet (as discussed in the next question), as this allows for less-obstructed passthrough of gradients between layers.)

   While their reasoning for this is sound, it is arguably a workaround and likely does not generalize to all neural network architectures. "Delving Deep into Rectifiers" attacked a deeper problem in the networks: that the rectifier activation functions used throughout the Inception network may not be optimally passing gradients through. The authors of this paper discovered that the initialization of filter weights when using rectifiers is important to how well the rectifier passes through the signal (i.e., how much a rectifier affects the standard deviation of the weights). By choosing the standard deviations of the randomly-initialized weights (rather than choosing a constant standard deviation), the authors are able to have a better translation of gradients through the network, which helps with training deeper networks (such as Inception).

2. *Explain the core effect of pre-activation from "Identity Mappings in Deep Residual Networks" compared to the original residual formulation.* (arXiv:1603.05027)

   The original ResNet formulation (from "Deep Residual Learning for Image Recognition" (arXiv:1512.03385)) used the idea that perhaps networks could learn better if they only had to learn the "residual" (which is somewhat like the error from the output) rather than try to learn the entire output. How this was implemented was by allowing "skip connections" between layers, which (roughly) passed through the gradient of the previous layer as well as that of the current layer – with this modification over a regular neural network, the difference caused by the current layer (the residual) can be learned. This helps train deeper networks faster, and also helps with learning identity layers (which may be useful in very deep networks).

   The paper "Identity Mappings in Deep Residual Networks" examines different variations of the original ResNet formulation. The most significant change they find is that if, rather than using an activation function after each ResNet block (i.e., after combining the output of the current layer and the previous layer), they created an asymmetric form (reordering) of the components of the ResNet such that the activation function acts as the first function in the ResNet block (see the following figure, taken from this paper).

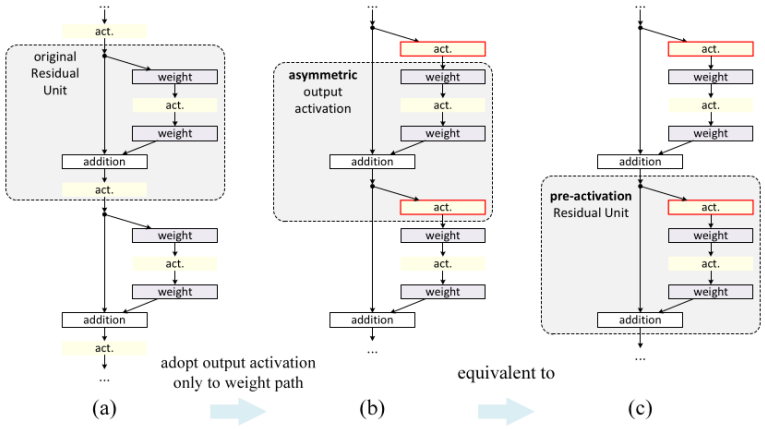


**Figure 5.** Using asymmetric after-addition activation is equivalent to constructing a *pre-activation* Residual Unit.

   By doing this, they create "paths" for each (modified) ResNet block that "branch off" and "rejoin" the "main path." Most of this is very similar to the original, except that now no activation functions were on the "main path." This means that the main path is (very simply) the sum of the original signal and all of the residuals (and the gradient is also a simple sum of the residual gradients). The authors reasoned that this allows gradients of earlier layers to much more easily flow between layers (as opposed to being "obstructed" by the activation functions on the main path), which allows for faster optimization (and also reduced overfitting according to the author's tests).