1.1
See Code: mlp/layers.py

1.2
GELU is defined by following equation

$$f(x) = 0.5x(1 + tanh[\sqrt{2/\pi}(x + 0.044715x^3)])$$

For derivative:

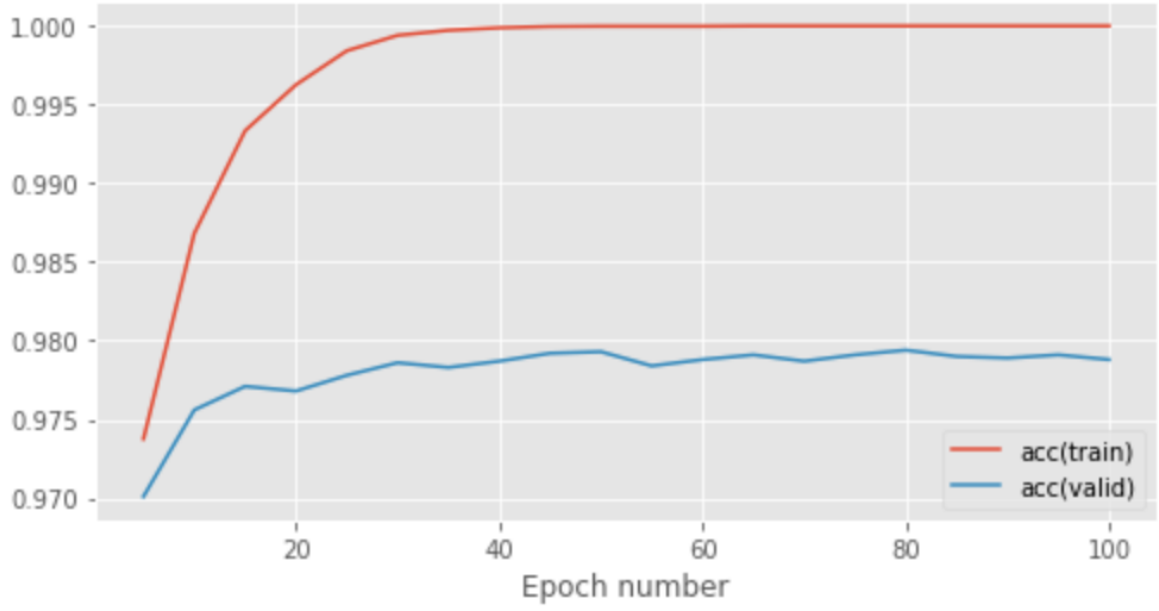
$$f'(x) = 0.5tanh(0.0356774x^3 + 0.797885x) + 0.5 + (0.0535161x^3 + 0.398942x) \times cosh^{-2}(0.0356774x^3 + 0.797885x)$$

Analysis and comparisons:

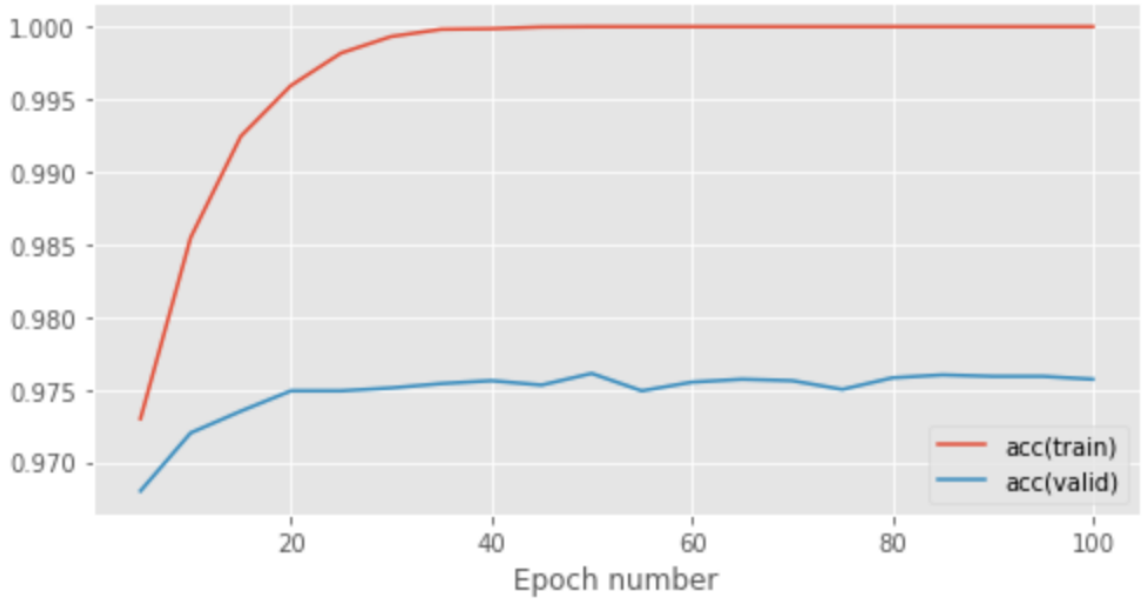
Initializations

I found out that lesser the initial weights faster the model was able to converge.
Did grid search based optimization for RELU (base model)

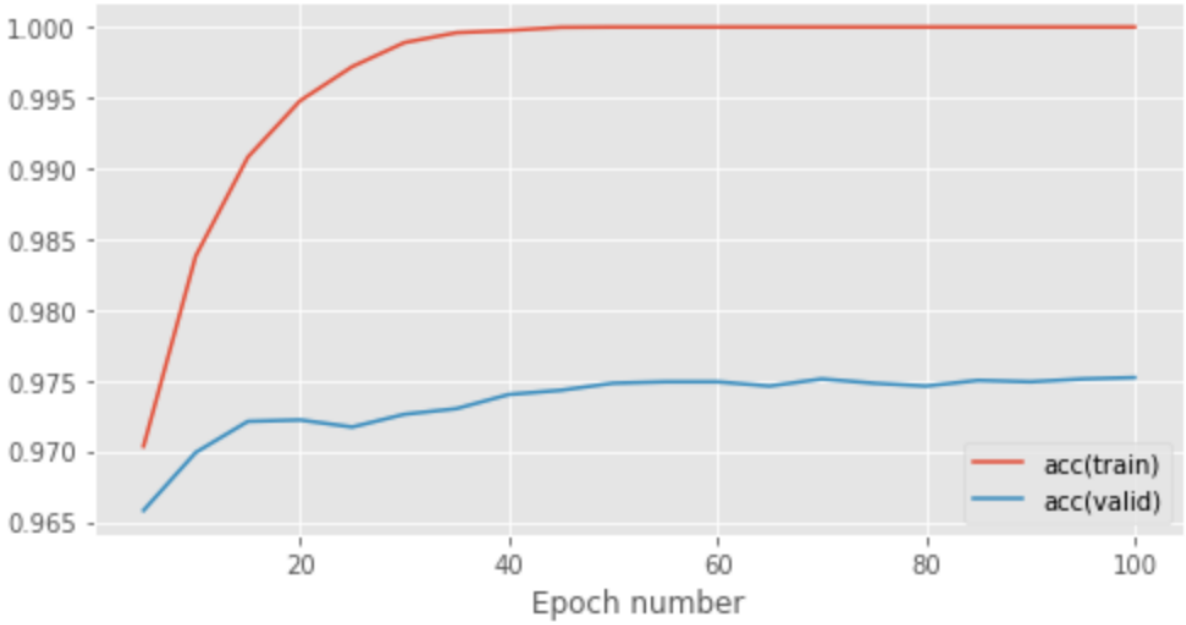
Relu 0.1 initial weight



Relu 0.2 initial weight



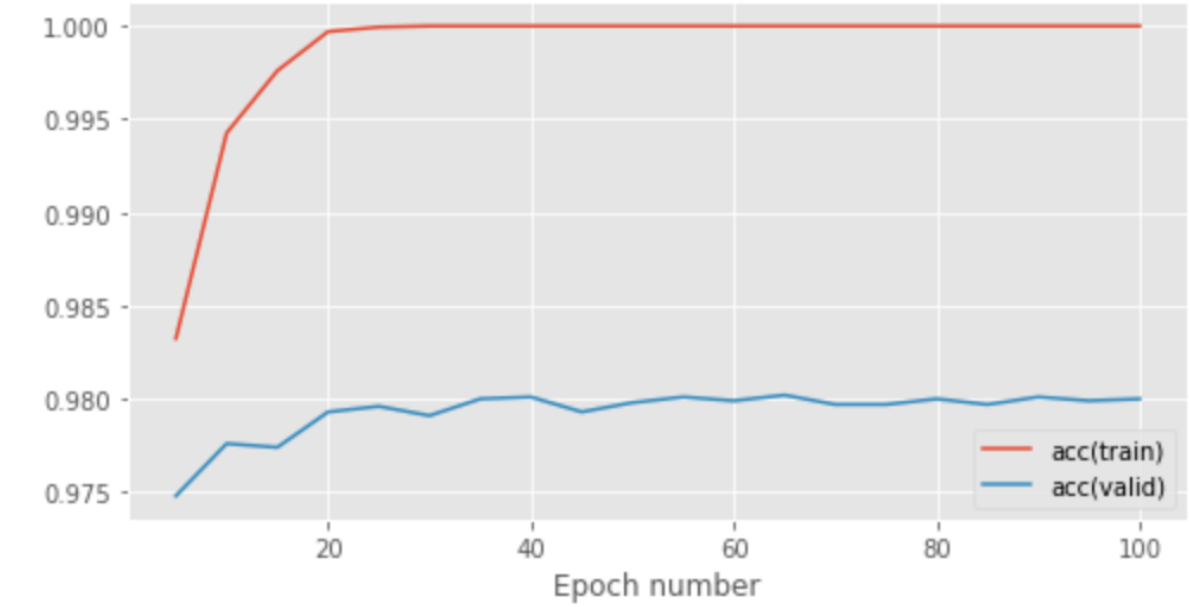
Relu 0.3 initial weight



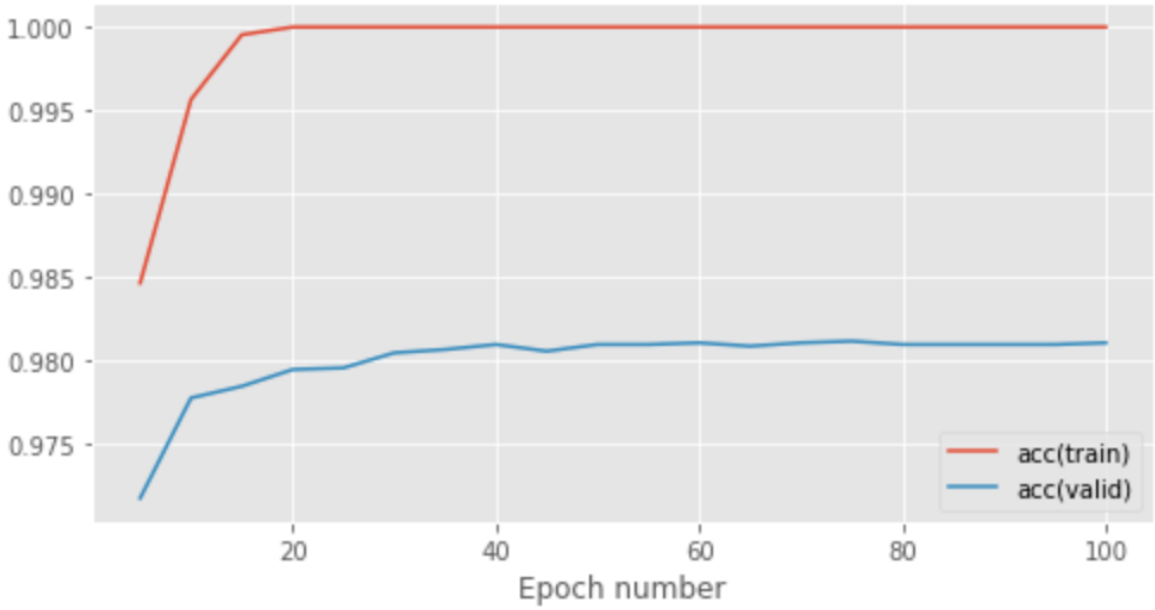
Learning Rate

More learning rate until 0.8 fetched faster convergence

RELU 0.4 learning rate

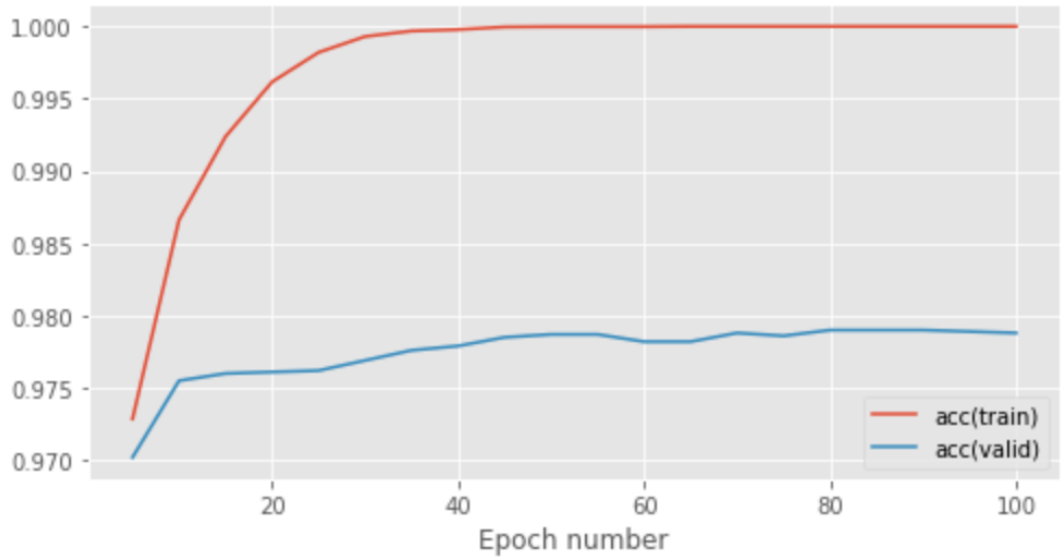


RELU 0.8 learning rate

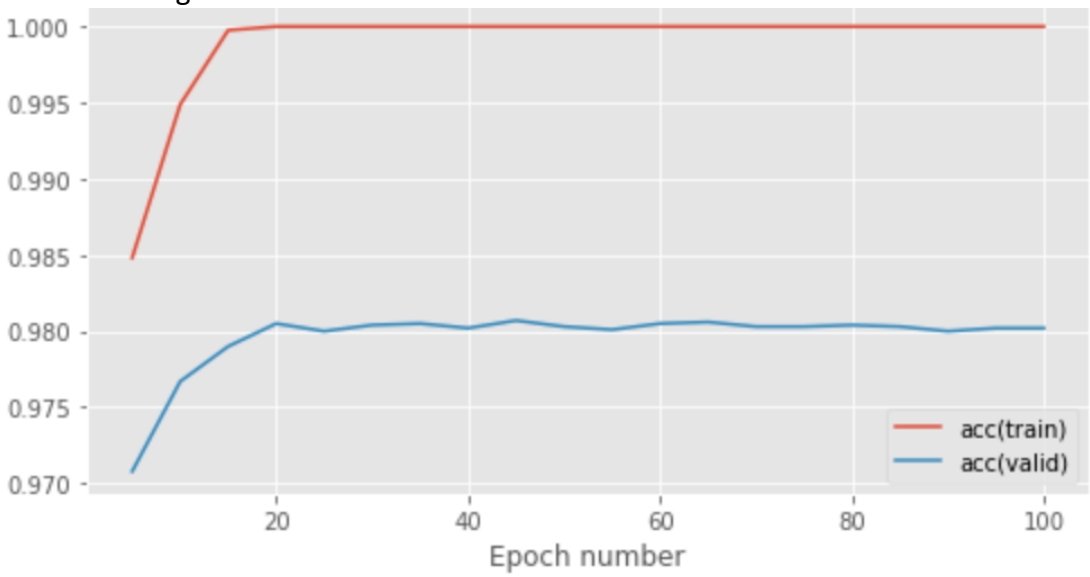


Wrt activation functions I found that using SELU model was not able to converge
GELU was the one with best (faster convergance and higher accuracy), ISRLU was more stable(not converging also)

GELU with learning rate 0.4



GELU with learning rate 0.8



Improvements:
Better pre-processing and grid search based optimizations will yield more improved results