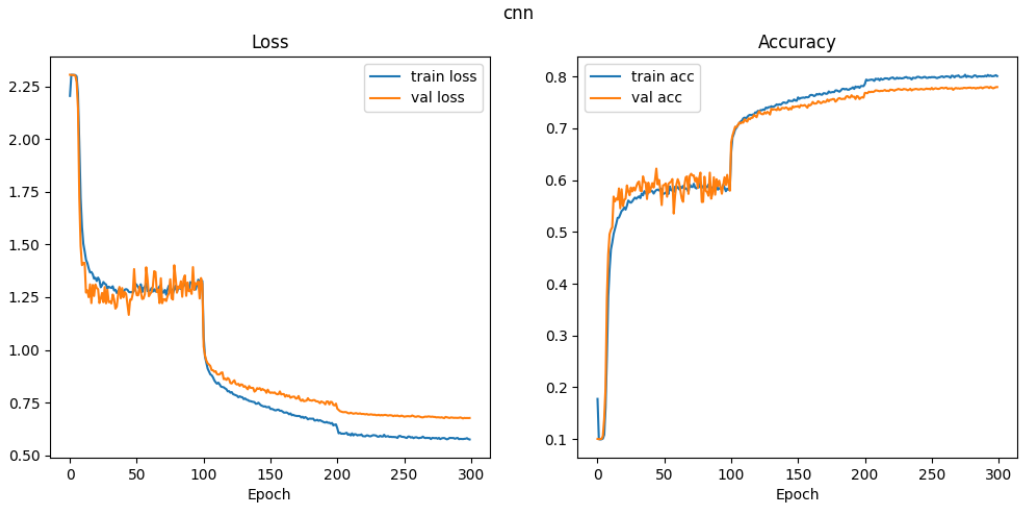
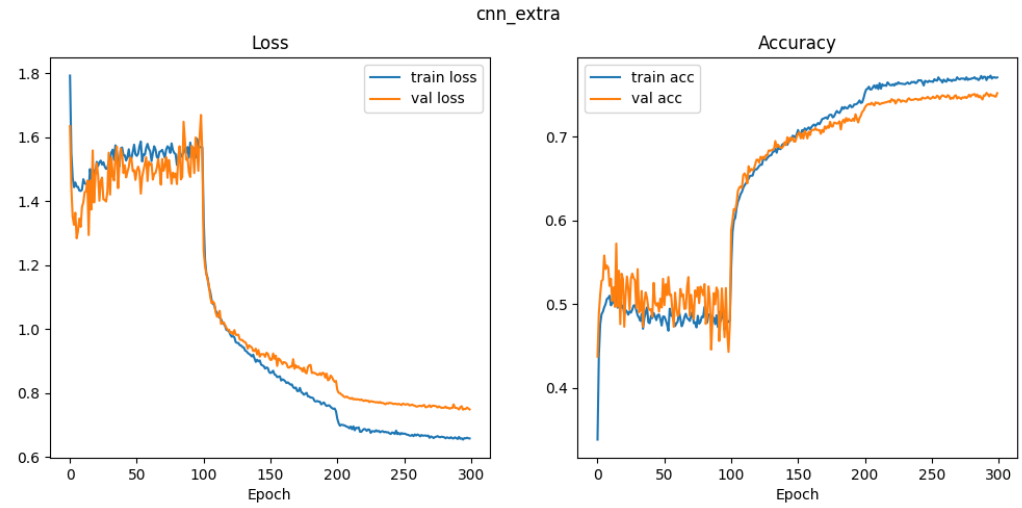
1a)



Model	Epochs	Best Val Acc	Final Val Acc	Final Val Loss	Params	Model Size (MB)	Total Time (s)
cnn	300	0.7801	0.7797	0.6772	1,070,794	4.085	486.890

This model had limited overfitting. It is considerably larger than the fully connected model that we trained for homework 2. As I have a good GPU, I was able to train all the models in parallel, but that was not possible for most of my peers.

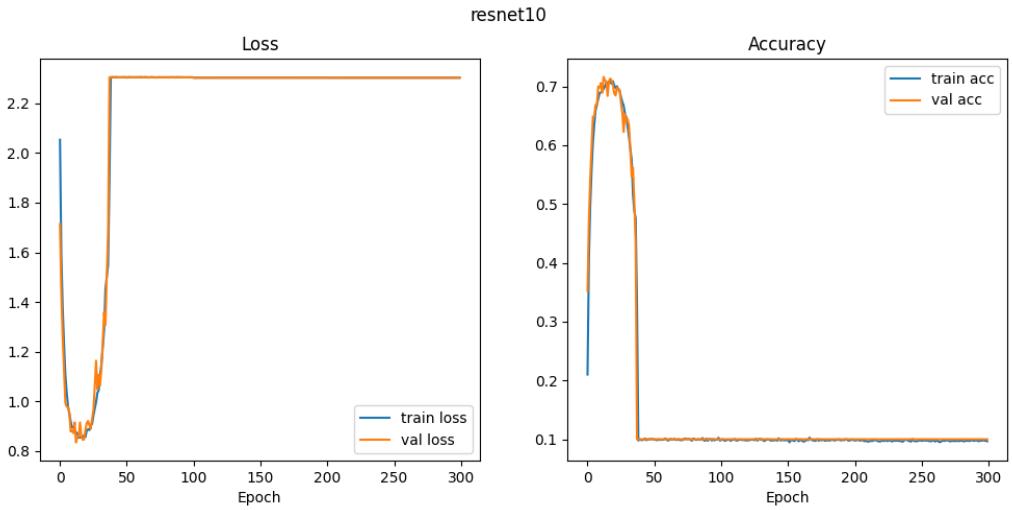
1b)



Model	Epochs	Best Val Acc	Final Val Acc	Final Val Loss	Params	Model Size (MB)	Total Time (s)
cnn_extra	300	0.7522	0.7519	0.7488	620,362	2.366	495.227

The accuracy is slightly worse than the baseline and has more overfitting than the baseline.

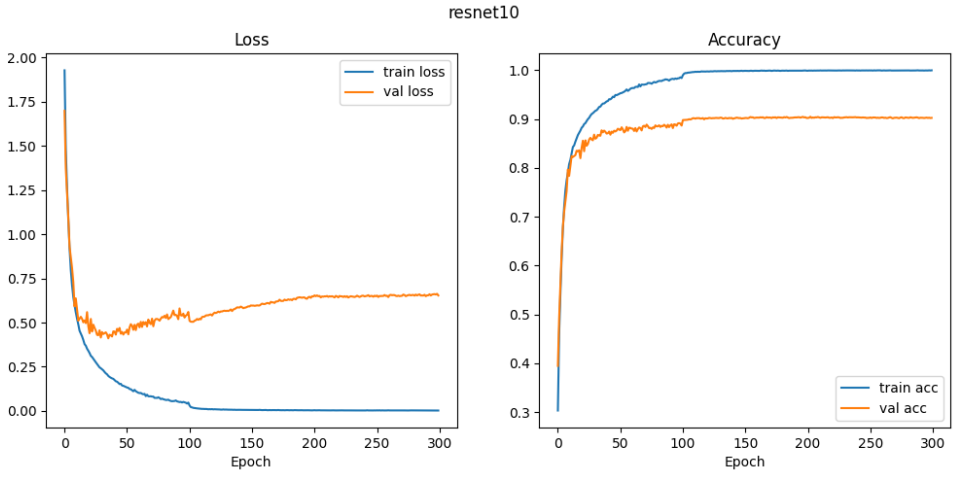
2) Resnet_baseline



For the baseline, it trained in considerably less than 300 epochs. It only needed 30 in order to train well. An accuracy of 70% is quite common in baseline models and good enough for this run.

Model	Epochs	Best Val Acc	Final Val Acc	Final Val Loss	Params	Model Size (MB)	Total Time (s)
resnet10	300	0.716	0.1	2.3025853931427	1,290,602	4.923	633.725

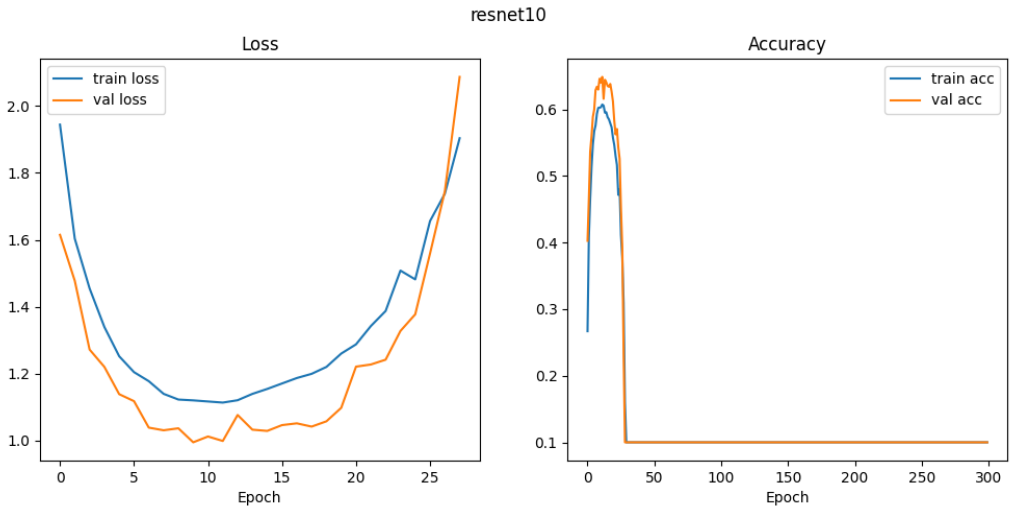
Resnet batch normalization



With batch normalization it was able to get to a roughly 85% accuracy before it started overfitting. It is better than the baseline.

Model	Epochs	Best Val Acc	Final Val Acc	Final Val Loss	Params	Model Size (MB)	Total Time (s)
resnet10	300	0.9045	0.9022	0.6538	1,292,314	4.930	814.446

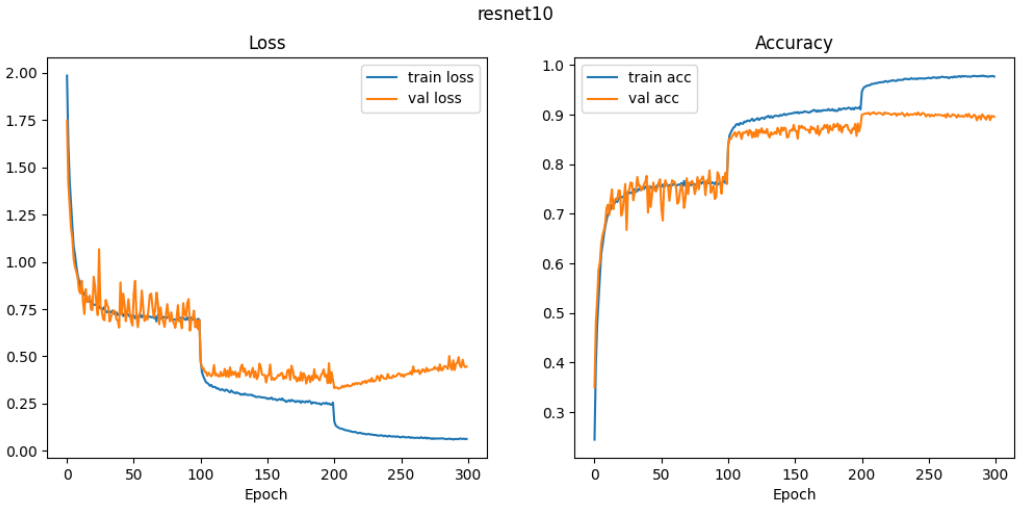
Resnet dropoout



With a dropout layer, I got an accuracy of 65% and it was worse than the baseline.

Model	Epochs	Best Val Acc	Final Val Acc	Final Val Loss	Params	Model Size (MB)	Total Time (s)
resnet10	300	0.6489	0.1	nan	1,290,602	4.923	674.435

Resnet Weight decay



With Weight decay, I was able to get an accuracy of 85% before it started overfitting. This is likely the best model. The correct solution would likely be a collection of all the approaches to get the most accurate model.

Model	Epochs	Best Val Acc	Final Val Acc	Final Val Loss	Params	Model Size (MB)	Total Time (s)
resnet10	300	0.9048	0.8958	0.4462	1,290,602	4.923	648.100