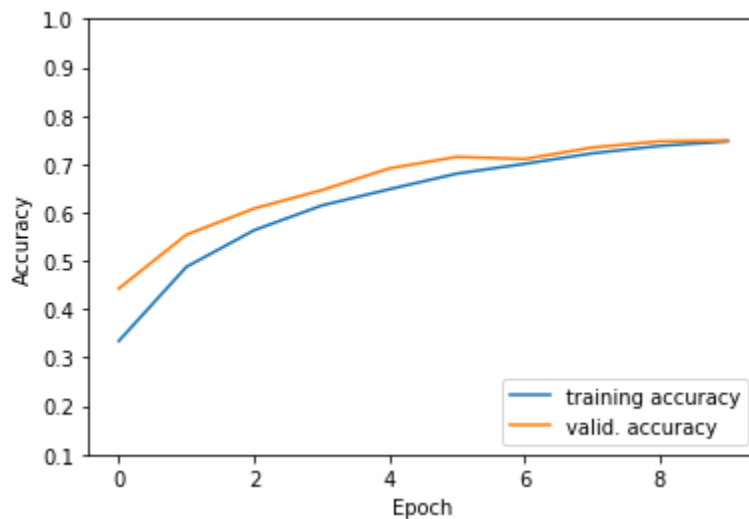


### Best Model(s):

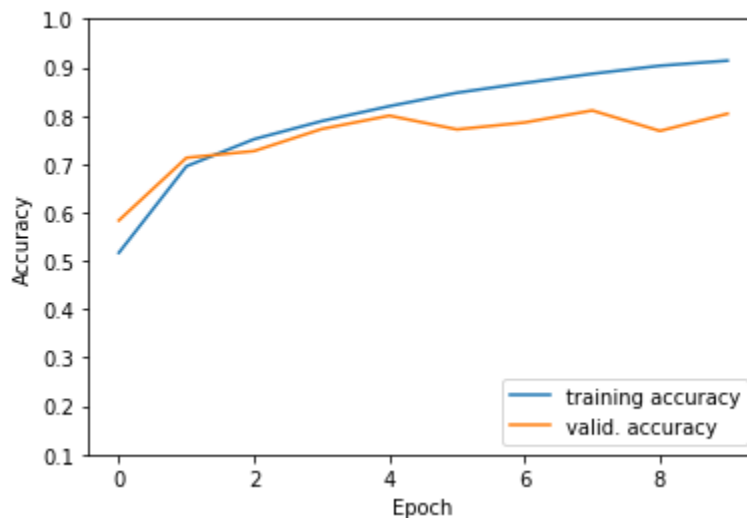
For this project, there are two models that offered good performance. The first is model 7 which displayed a final accuracy of 74.86% and appears to be a good fit for the data as the plot indicates with the training and validation data:

`valid_accuracy=0.7486000061035156, valid_loss=0.7078680396080017`



The second is model 10, which returned a final accuracy of 80%. There seems to be some slight overfitting, but this may be overcome by using more Epochs as there appears to be a pattern in the plot.

`valid_accuracy=0.8040000200271606, valid_loss=0.626409649848938`



Model 7 was constructed using the ‘Adam’ optimizer, several layers, Batch Normalization, Padding and Spatial Dropout. It should be noted that Batch Normalization was performed after the layers were added and after dropout. This was something that I experimented with back and forth and could not determine if there is a more preferred method to apply the Batch Normalization.

Model 10 was constructed applying Batch Normalization to each layer and using Dropout and NOT Spatial Dropout. Padding was also used as well as the ‘Adam’ optimizer.

### **Description of Journey:**

The initial path was to build on top of the first given model. Small steps were taken as the layers were adjusted from model to model. The plan was to adjust the layers until the best performance was found and then adjust the given parameters such as optimization and introduce parameters such as padding and regularization. The filters were “touched”, but this resulted in poor performance and they were ultimately returned to the initial size. The number of Epochs was to be randomly increased from model to model. Unfortunately, this option was scratched as the more the parameters were tuned, the longer it takes for the Epochs to run. While the initial model viewed ‘accuracy’ as its metrics, the plan was to experiment with other metrics or maybe include the models with the lowest loss as a factor in determining the best model.

The models were built in the Anaconda PC environment using Jupyter notebooks. For the first few models there were no attempts made to adjust the GPUs. This could explain the lengthy time for the Epochs. Any model over 20 Epochs ran for an hour or longer. What is an acceptable run-time?

Building on top of the initial model, performance seemed to be steady with the adjustment of layers. Performance dipped significantly with the addition of the parameters. This proved to be quite a bit of trial and error with the execution of an untold number of models. The models in the notebook are the initial benchmarks of their best performance before parameter adjustment.

### **Final conclusions:**

After building on each model and looking at its performance (using accuracy), the effect of tuning the parameters and adjusting different layers can be observed. This can explain how and why there is no one-size-fits-all model, or neural network. In hindsight, for this project, the data should have been reviewed, as it is possible that it could have had a negative impact on the model performance as well. One question that I do have is “For this data, what does the optimal model look like?” Also, in the case of neural networks and this data set, what is an acceptable accuracy percentage, or should an acceptable model include a number of factors? Is it possible to have a perfect model?

## **Reaction and Reflection**

I expected this project to be challenging. The main issue that I have is the “Kaggle” score. What I don’t understand is how the models produced accuracy rates that were in the 74 – 80% range, yet the “Kaggle” score indicates that they may be very poor performing models. This is beyond perplexing. I was very tempted to upload the “given” submissions.csv file to see if there was a local issue (my P.C., code or connection) that caused my scores to be low. Also, I attempted to populate the test\_labels[] array, but encountered an issue with that.

The next issue that I had was working with Tensorflow and Keras. This course is my first experience with both. After reviewing the documentation, I should have spent more time applying and experimenting with the parameters. Then reviewing again what each parameter does and its overall impact on the model. I’ve found several online tutorials that I plan to complete at the conclusion of this course.