1. Your objective should be stated in written form. What are you trying to accomplish? Predicting a number? Classifying? Your objective must reference the context of the problem, specifically. (You may reuse the objective from Deliverable 1, or revise as appropriate.)

My objective is classifying the avocado category (organic or conventional) by using 12 features given by the dataset includes features like date, total volume, total bags, small bags. My metric is test accuracy, calculated by how many correct test cases the model predict divided by the total test case

1. The source URL(s) for the data and description should be included.

<https://www.kaggle.com/neuromusic/avocado-prices>

The data contain 14 column and 18250 row. For each column definition, you can refer part 7

1. Your final ANN model, in code, in an attachment.
2. Your final model and training algorithm, in words.

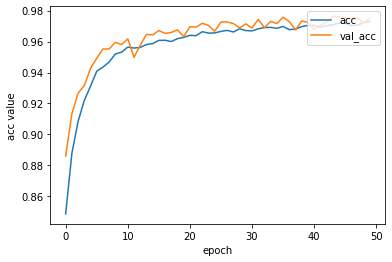
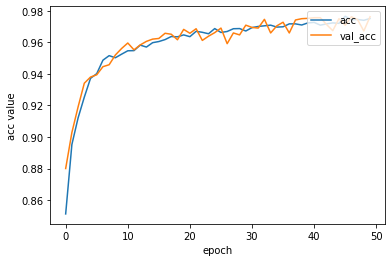
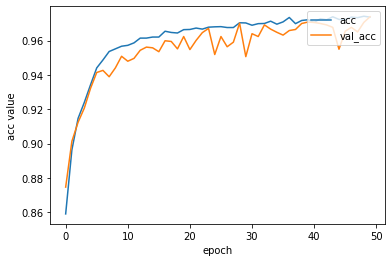
**The final model I create has 3 layers. The input layer has 256 hidden units with activation function relu, the hidden layer has 256 hidden units with activation function relu, and the final layer has one hidden unit with activation function sigmoid. Between each layer, I set a 10 percent dropout rate. The optimizer I choose is adam, with learning rate 0.01, other parameters are default setting. The loss is binary\_crossentrophy and the metric is test accuracy. The epoch is 150 and batch size is 10. I use a 20 percent of training data as validation data to track the validation loss and validation accuracy. I also use callback to find the best epoch and use that weight to fit my test data**

1. Your experimental plan for arriving at the final model.

**I build 3 extra models before I arrive the final model.**

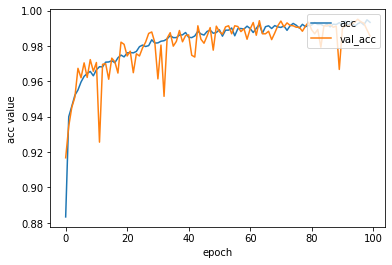
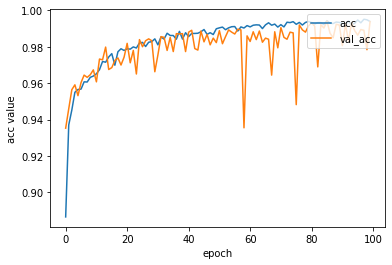
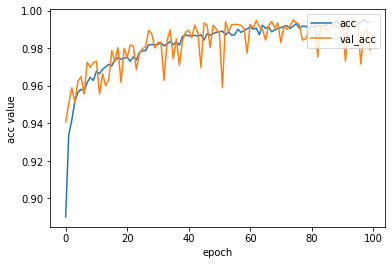
**The first one is a base model with 2 layer. 64 hidden units with relu in the input layer and 1 hidden unit with sigmoid in the output layer. I use default adam optimize algorithm and 50 epoch, 10 batch size, and use 3 fold validation. The loss and metric is same as the final model.**

**The val\_acc, train\_acc plot by epochs for each cv model:**

**  **

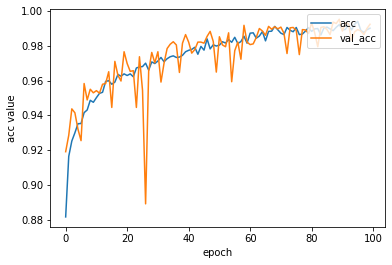
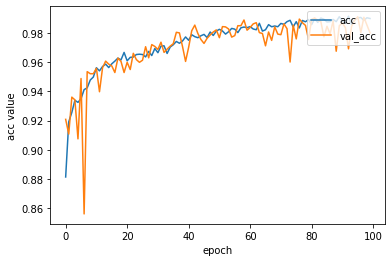
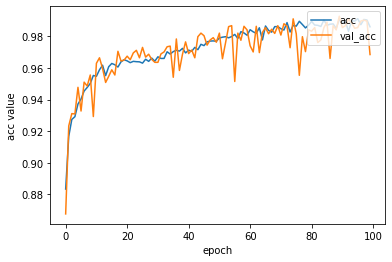
**The second one is a 5 layer ANN. 64 hidden units with relu in the input layer, 3 layers in hidden layer with relu and 128 hidden units, 1 hidden unit with sigmoid in the output layer. I use default adam optimize algorithm and 50 epoch, 10 batch size, and use 3 fold validation. The loss and metric is same as the final model.**

**The val\_acc, train\_acc plot by epochs for each cv model:**

**  **

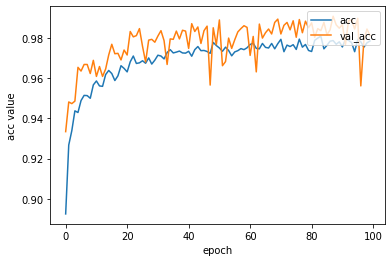
**The third model is also a 5 layer ANN, I use tanh as the activation function for the input layers and all hidden layers. The input layer has 64 hidden units while the hidden layers have 128 hidden units for each layer. The output layer has one hidden unit and sigmoid as the activation function. I use default adam optimize algorithm and 50 epoch, 10 batch size, and use 3 fold validation. The loss and metric is same as the final model.**

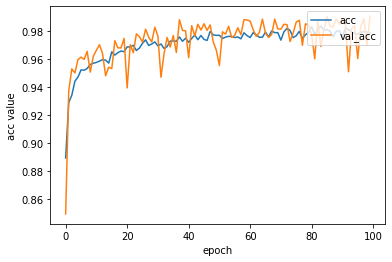
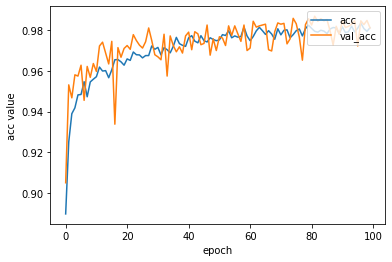
**The val\_acc, train\_acc plot by epochs for each cv model:**

**  **

**Even though all 3 model has relatively good result, over 96% validation accuracy, but from the graph I find the validation accuracy is oscillating. So my final model add dropout and change the learning rate from 0.001to 0.01 in order to have more flat validation accuracy history. I also use callbacks to find global optimal weight for my epochs.**

**The val\_acc, train\_acc plot by epochs for each cv model:**

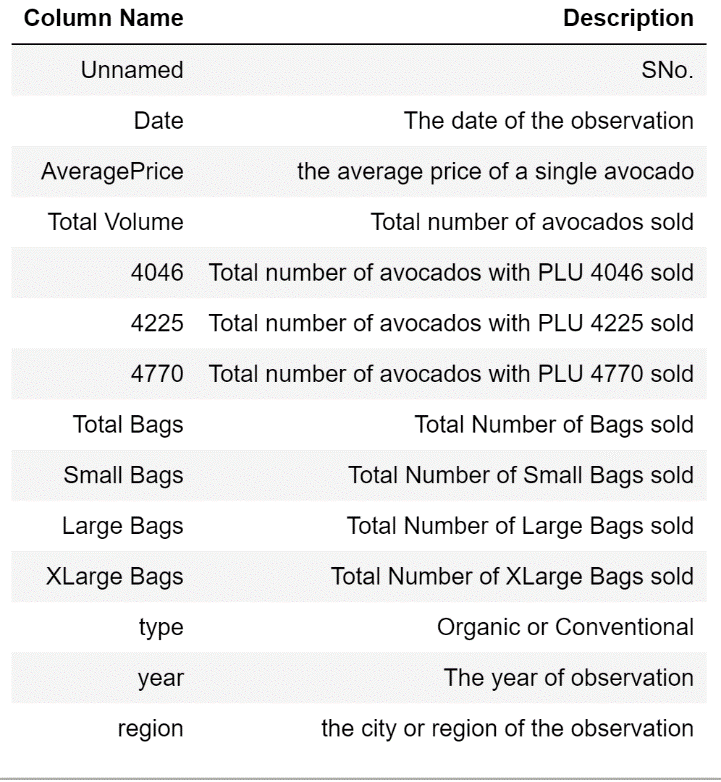
****

1. How long it took to run all the models in your experimental plan.

**2 hours in Google colab TPU**

1. An explanation of the input variables



1. The data preprocessing steps you took. And why you took those particular steps
   1. **Drop the index column, which is unnecessary for fitting model**
   2. **Change the 4046,4225,4770 column to more understandable column name**
   3. **Change the type to dummy variable**
   4. **Label encode region. ANN can’t take string variable**
   5. **Delete date, extract month in date and transfer month to season, reduce category for better model performance**
   6. **Drop total volume, total bags because they are sum of other variables.**
   7. **Scale all the features except type, month and year for better model performance**
   8. **Encode year since 2015 is too large for fitting model**
   9. **Split the data 80/20 in train and test**
2. An explanation of your metrics and justification for your choice.

**I use test accuracy as the metric. This is straightforward for binary classification. You can know how many correct test cases you classify clearly by compare with the actual test cases. The reason why I don’t use precision/recall is that I think test accuracy is more straightforward and understandable**

1. An explanation of your method to validate the model.

**I use 3-fold cv to validate my model. I randomly divide my training data to 3-fold. I use 2-fold as training set to train the experimental model and the rest fold to validate the model. For each fold, it has the chance to be training set and validation set. Then I calculate the mean and standard deviation for all 3 cross validation model to have a less biased result. In that case, I will not overfit and yield a biased result especially ANN has huge randomness.**

1. Your results in terms of appropriate metrics for the objective and problem.

**98.767% test accuracy for classifying the avocado type**

1. A discussion and/or justification for how you used/didn’t use all of the following:
   1. selection of the optimum number of units

**During experiment, I find the larger hidden unit I choose, I will have better validation accuracy, so I change my units from 64 to 256.**

* 1. type of network (feedforward, recurrent, backpropagation, and etc.),

**My network is backpropagation, by initialize empty weight and change to optimal after enough iteration.**

* 1. type of training (supervised, unsupervised),

**Supervised. I have labels and use labels in test case to build metric—test accuracy.**

* 1. proportion of training and testing data sets (70:30, 80:20, and etc.),

**I use 80:20, because my dataset does not have enough instances for training the network (less than 19000 instance). Network needs larger dataset to yield good result.**

* 1. number of input and output units (usually application dependent),

**10 input units and 1 output units. I have 10 features feed into neural network.**

* 1. number and size of hidden layers (2N+1, experimental),

**I try 3,1 and 0 number of hidden layers and find out the number of hidden layers will not cause significant difference**

* 1. number of repetitions during training (epoch),

**150 epoch. Initially I use 50 epoch in base model but I find the loss curve is still decreasing.**

* 1. choice of activation function (sigmoid, linear, Tanh, ReLU, and etc.) and

**I use sigmoid in output layer for all models because this is a binary classification problem, other activation function like softmax will not yield good result. I prefer relu for input and hidden because the model yield better result when using relu than tanh**

* 1. size of data set (number of records)

**18250 rows in total**

* 1. learning rate

**My final model use 0.01 as learning rate, because the experimental plan shows oscillating validation accuracy, and the initial validation accuracy is high.**

* 1. momentum.

**I use the default momentum in adam. I am satisfied the learning pattern, there is no need for change momentum, then result does not take too long to converge.**

1. Discussion of results and further work.

**The result is good, 98.767% test accuracy, I may need use data augmentation to bring more instance to train the model in order to have better accuracy. The instance is not enough for feed the model .**

1. For each line of the code that you used for the assignment, other than those containing ‘from’ and/or ‘import’, please insert a comment above stating what each line does.
   1. Please place a comment block (code flow) below the ‘from/import’ block and above the code that describes in sentence form the overall flow of the code and the purpose of the code (what are we trying to accomplish?). You may use pseudocode, if desired.
   2. Please place a comment block below the ‘code flow’ block and above the code that lists each variable used and explains what each variable is used for.
   3. A single line comment may start with a #.
   4. A comment block should start with ‘’’ and end with ‘’’.
   5. If you are commenting two or more lines of code that are essentially identical, i.e. different variables but identical operations, you may use one comment above that code block. For code that has similar operations (model.add), but has different parameters, please comment each line.
   6. Please do not combine comment blocks.