```
In [ ]:
```

Answer [CM3]

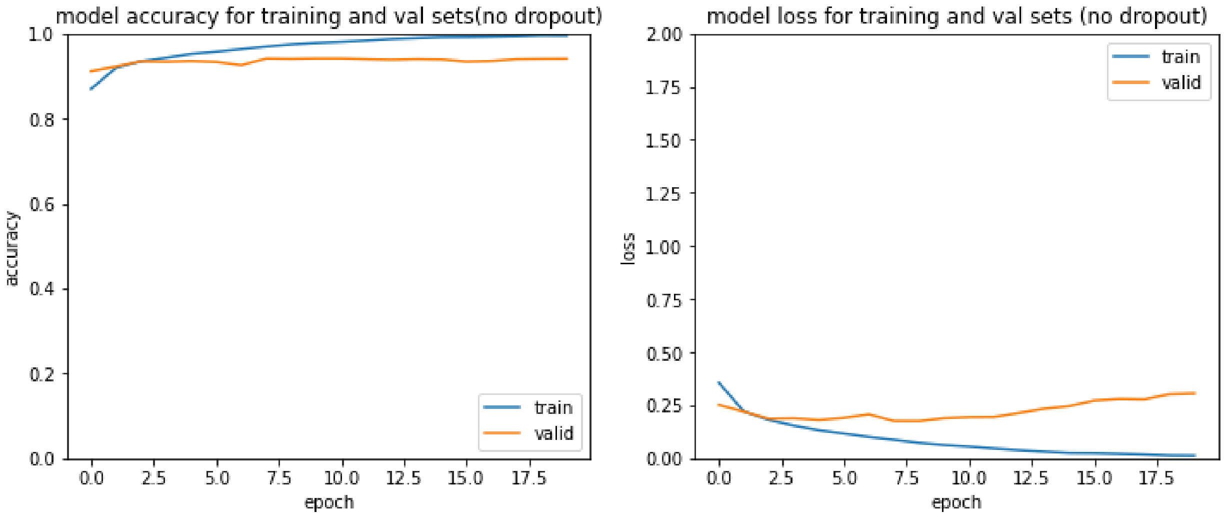
[CM1] and [CM2] results Analysis

1) Default Model

Plotting graphs of training & Validation loss vs. epoch and training & validation accuracy vs. epoch

```
In [32]: hist_d = train_model_default.history
acc_d= hist_d['accuracy']
val_acc_d = hist_d['val_accuracy']
loss_d= hist_d['loss']
val_loss_d= hist_d['val_loss']
epochs = list(range(1,len(acc_d)+1))

plt.figure(figsize=(12,10))
plt.subplot(2,2,1)
plt.plot(acc_d)
plt.plot(val_acc_d)
plt.ylim(0, 1)
plt.title('model accuracy for training and val sets(no dropout)')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'valid'], loc='lower right')
plt.subplot(2,2,2)
plt.plot(loss_d)
plt.plot(val_loss_d)
plt.title('model loss for training and val sets (no dropout)')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'valid'], loc='upper right')
plt.ylim([0,2])
plt.show()
```



From above graphs, we can say that

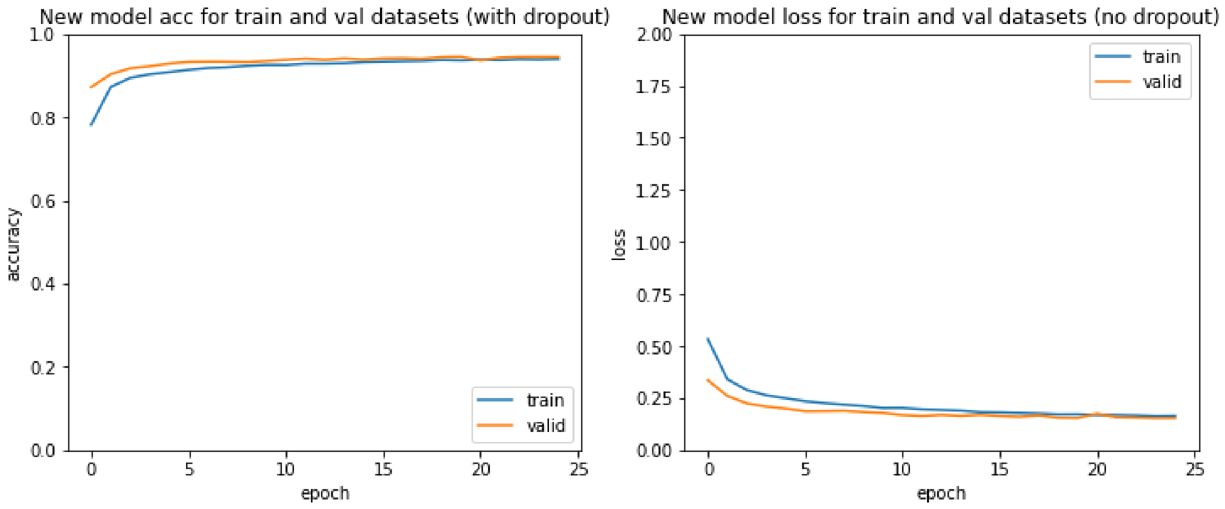
- Model is leading towards overfitting after 5 epochs(as training loss is decreasing while validation loss is slightly increasing afterwards)
- This can affect the testing accuracy and it will not perform well if we will try to train it will more number of epochs.

2) New model

Plotting graphs of training & Validation loss vs. epoch and training & validation accuracy vs. epoch of Our Own Network

In [33]:

```
hist = train_new_model.history
acc = hist['accuracy']
val_acc = hist['val_accuracy']
loss = hist['loss']
val_loss = hist['val_loss']
epochs = list(range(1,len(acc)+1))
plt.figure(figsize=(12,10))
plt.subplot(2,2,1)
plt.plot(acc)
plt.plot(val_acc)
plt.ylim(0, 1)
plt.title('New model acc for train and val datasets (with dropout)')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'valid'], loc='lower right')
plt.subplot(2,2,2)
plt.plot(loss)
plt.plot(val_loss)
plt.title('New model loss for train and val datasets (no dropout)')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'valid'], loc='upper right')
plt.ylim([0,2])
plt.show()
```



From above graphs, we can say that,

- The model is not gettting overfit even after running it to 25 epochs and the trends for train loss and valid loss are better than the previous model.
- Plus training accuracy is setting to one point and it is not fluctuating much after a certain epoch. By this we can interprete that if we will have more data or better labelled data, the model can perform better for more number of epochs.

In [37]:

```
y_pred = model.predict(X_test, batch_size=64, verbose=1)
y_pred_bool = np.argmax(y_pred, axis=1)

print(classification_report(test_data.label, y_pred_bool))
```

```
157/157 [==============================] - 1s 5ms/step
              precision    recall  f1-score   support

           0       0.89      0.91      0.90      2000
           1       0.99      0.97      0.98      1000
           2       0.93      0.93      0.93      2000
           3       0.97      0.98      0.97      2000
           4       0.94      0.93      0.93      3000

    accuracy                           0.94     10000
   macro avg       0.94      0.94      0.94     10000
weighted avg       0.94      0.94      0.94     10000
```

From the classification report of the new model, we cna observe that class 1 and 3 is performing very well. So it could be well labelled than the other classes.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: