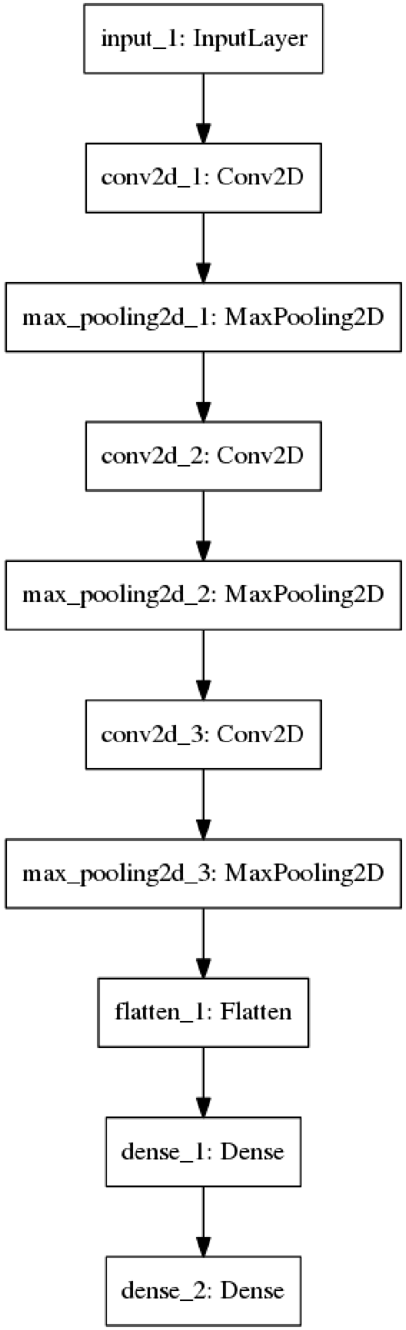
Face Recognition

Шахин Зейн
13546/2

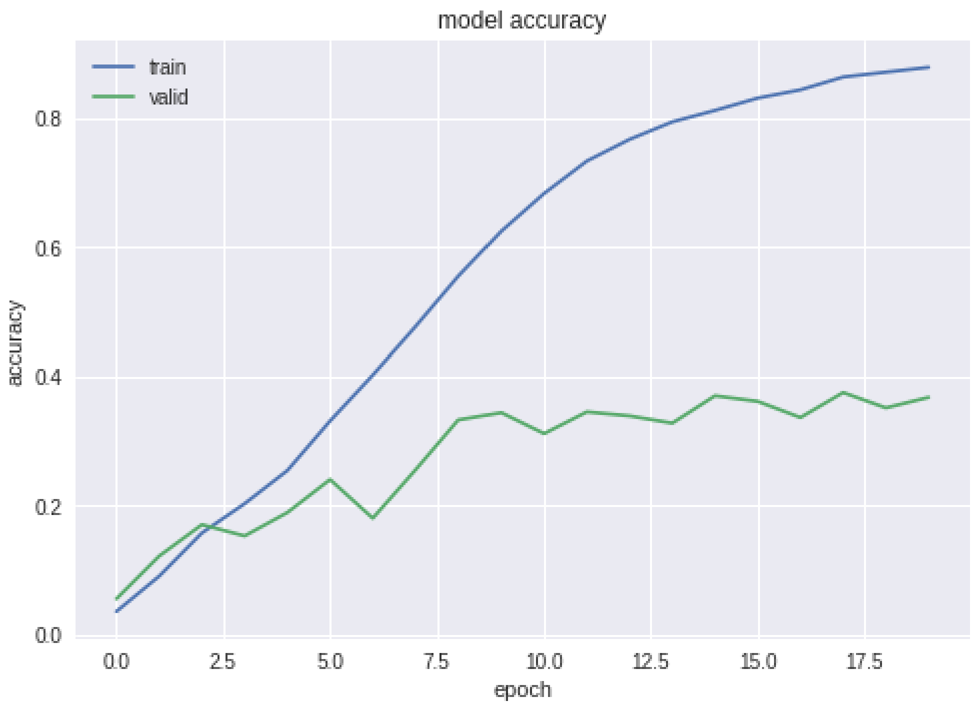
- The following presentation contains the results of my work.
- I tried to use different techniques with the baseline model proposed by the Lecturer.
  - Dropout with different values for dropout parameter.
  - L2 regularization with different values for regularization parameter.
  - Batch normalization.
- I tried to keep the structure of the network and not make much changes.
  - Only one experiment, I increased neurons of dense_1 to 128.
- The complete notebooks could be found on my repository
  - https://github.com/zeinsh/experementaldataprocessing/tree/master/FR

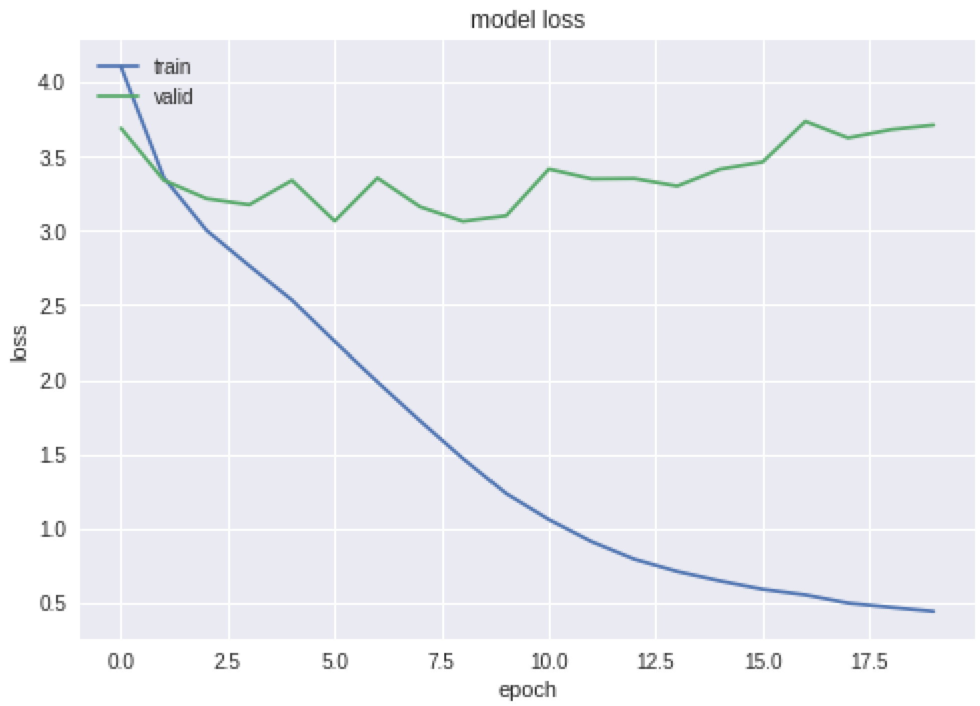


Baseline Model

- Network Structure
  - InputLayer 1x150x150
  - Convolution layers
    - Conv2d_1 32filters, size 3x3, stride 1x1
    - Conv2d_1 32filters, size 3x3, stride 1x1
    - Conv2d_1 64filters, size 3x3, stride 1x1
  - Max pooling layer
    - All pooling layers of size 2x2
  - Dense Layers
    - Dense_1 Dense(64)
    - Output layer Dense(83)
- Optimizer: Adam
- Loss function: sparse_categorical_crossentropy
- Quality Metric: Accuracy

Baseline Model





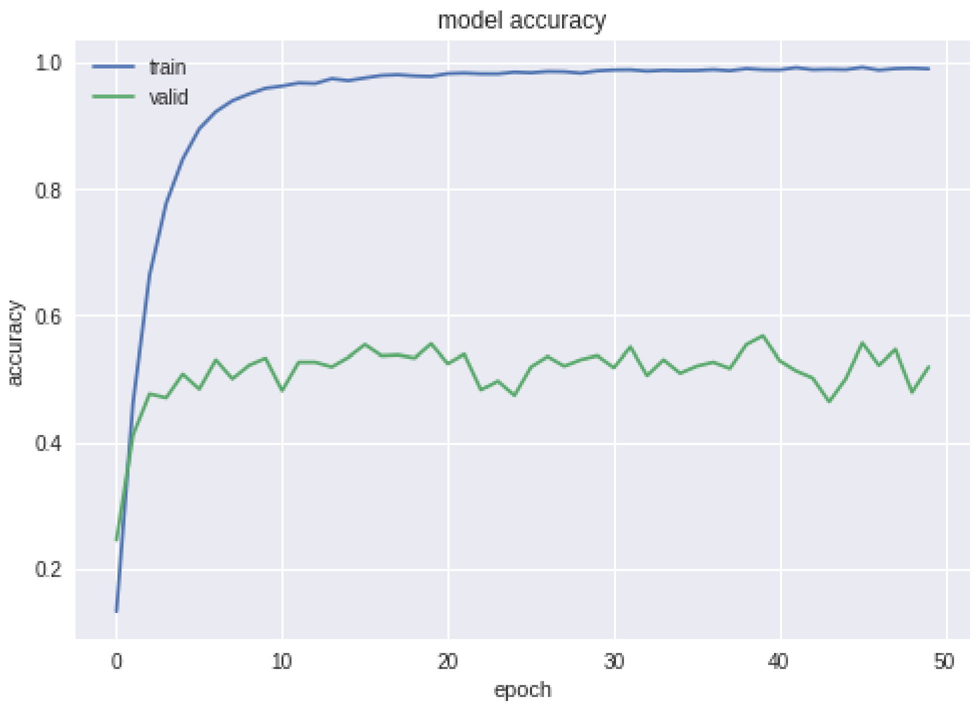
Baseline Model

- <u>Comments on this model</u>
  - There is overfitting, there are two approaches to reduce overfitting
    - Dropout
    - L2 regularization
  - Validation accuracy
    - The best value is 37%

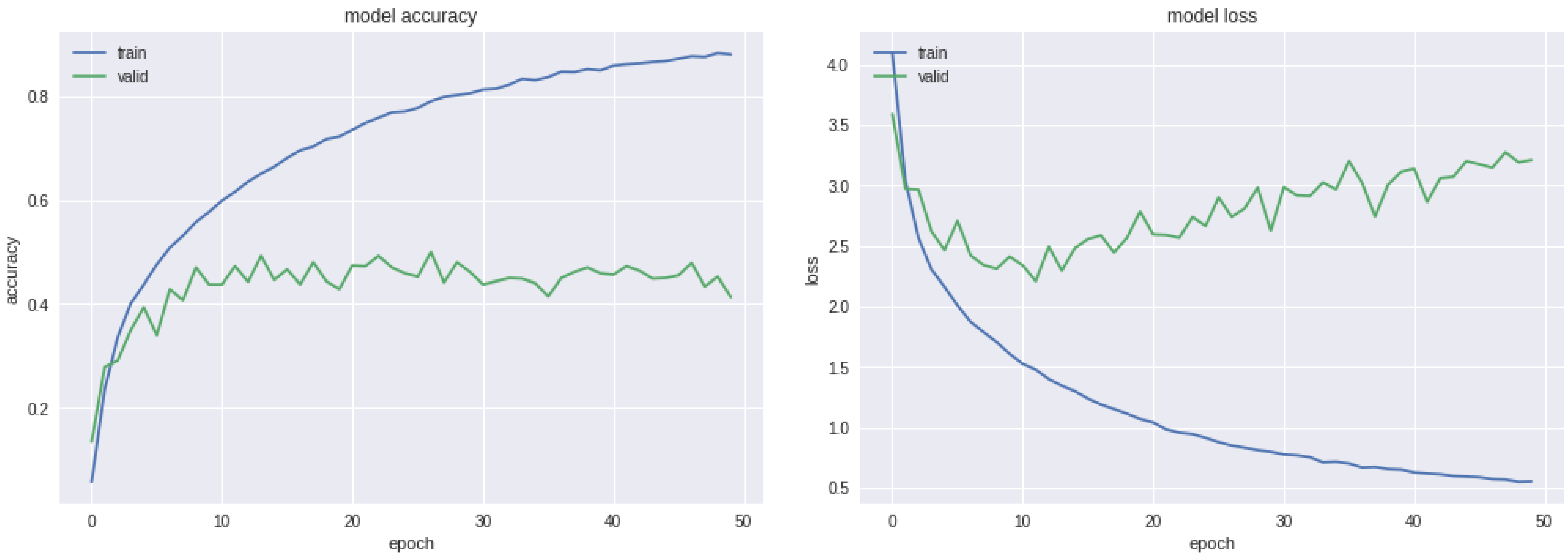
L2 Regularization

- **Add L2 regularization to convolution layers**
  - Using keras.regularizers.l2(l2_norm)
- Use these values for l2_norm
  - 0 : no regularization (Baseline)
  - 0.001
  - 0.005
  - 0.01

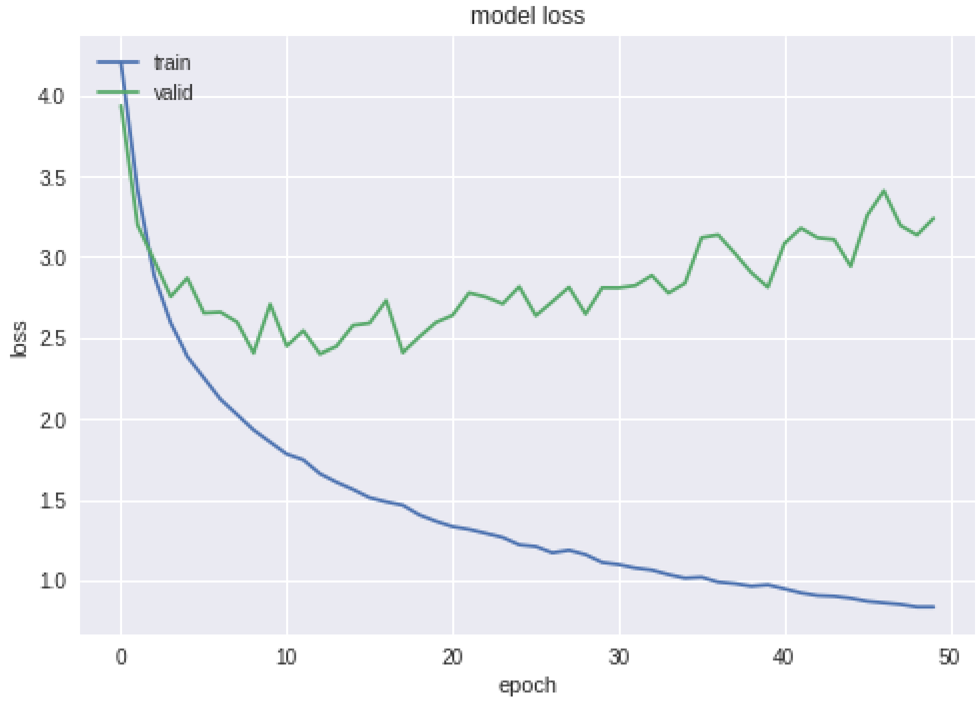
L2 regularization 0.001



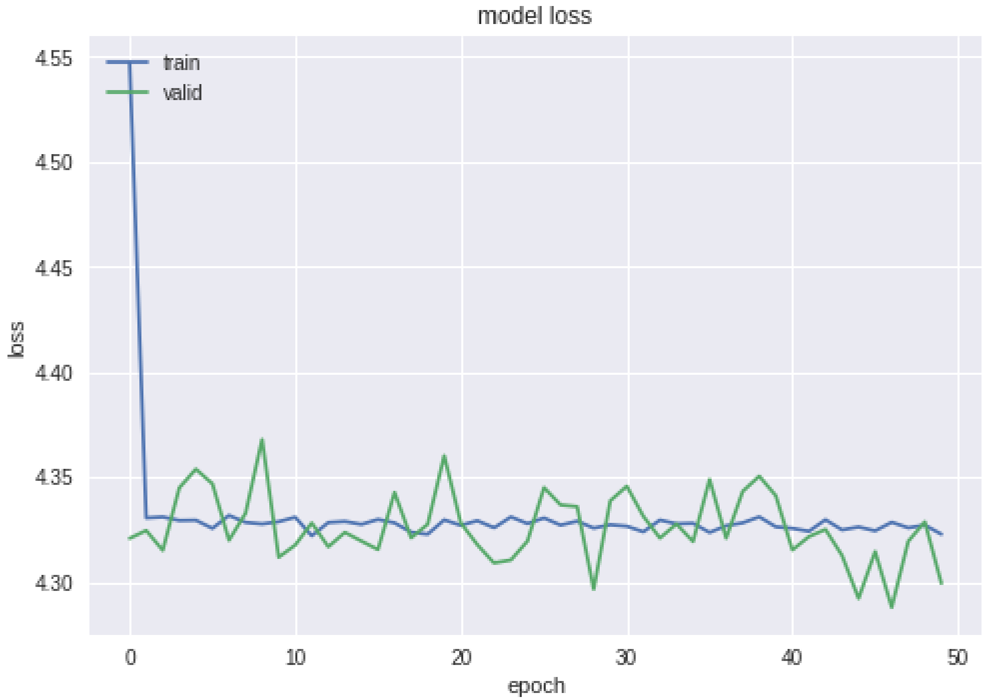
L2 regularization 0.005



L2 regularization 0.01



L2 regularization 0.1



L2 Regularization

- Comments on using L2 Regularization with baseline model
  - Increasing the regularization parameter doesn’t improve the model too much, but cause the validation accuracy to go down,
  - Best validation accuracy using L2 regularization

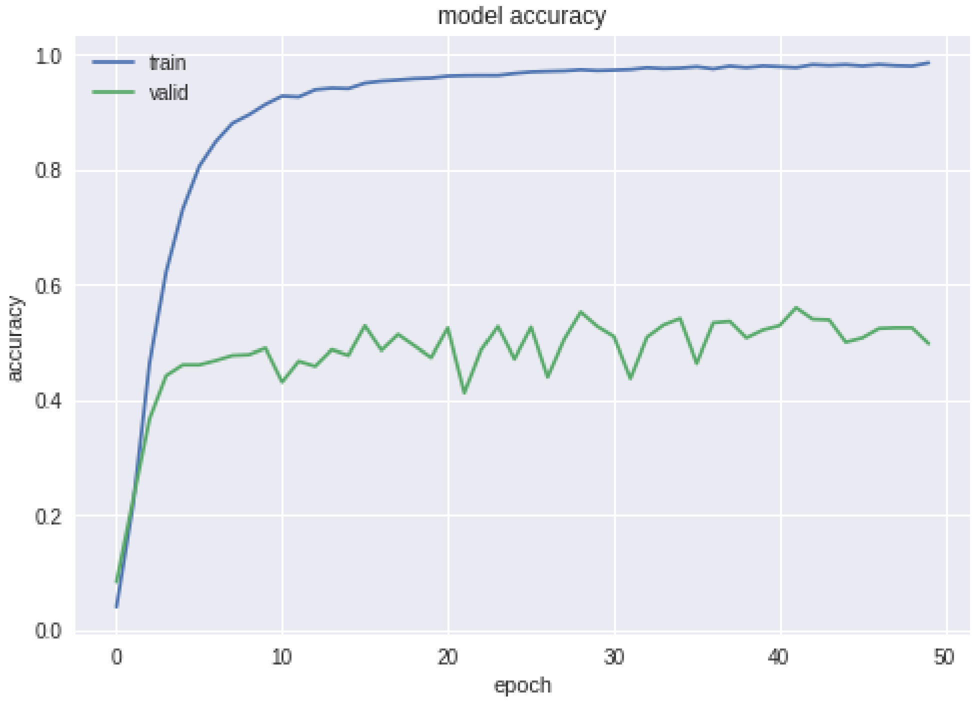
C	Best validation accuracy
0.001	55.37%
0.005	50%
0.01	44.25%
0.1	<1%

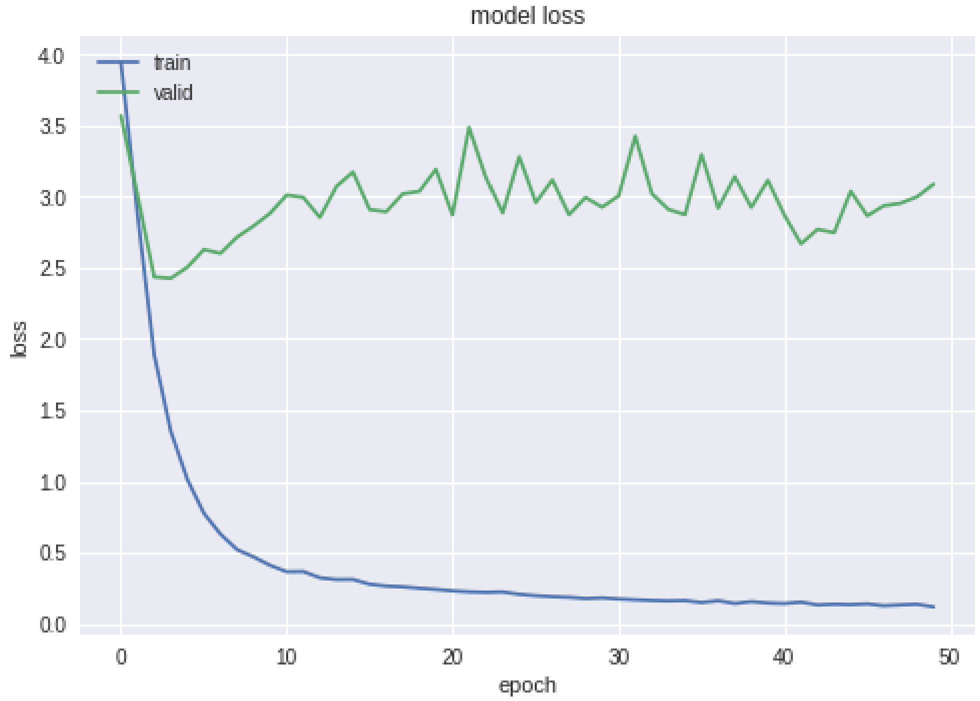
Increasing regularization parameter to 0.1 will cause to decrease network weights to very small values, though the model will learn nothing

Use L2 regularization with Batch Normalization

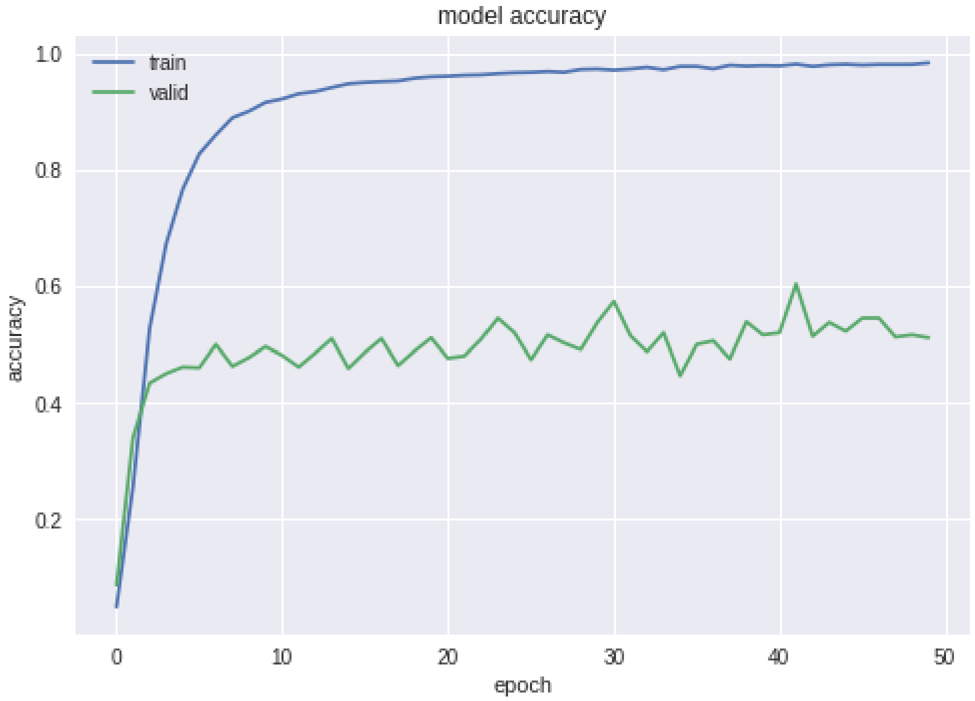
- Add batch normalization along the first axis (dimension related to channels)
- Use the previous values of l2_norm with Batch Normalization
  - 0
  - 0.001
  - 0.005
  - 0.01

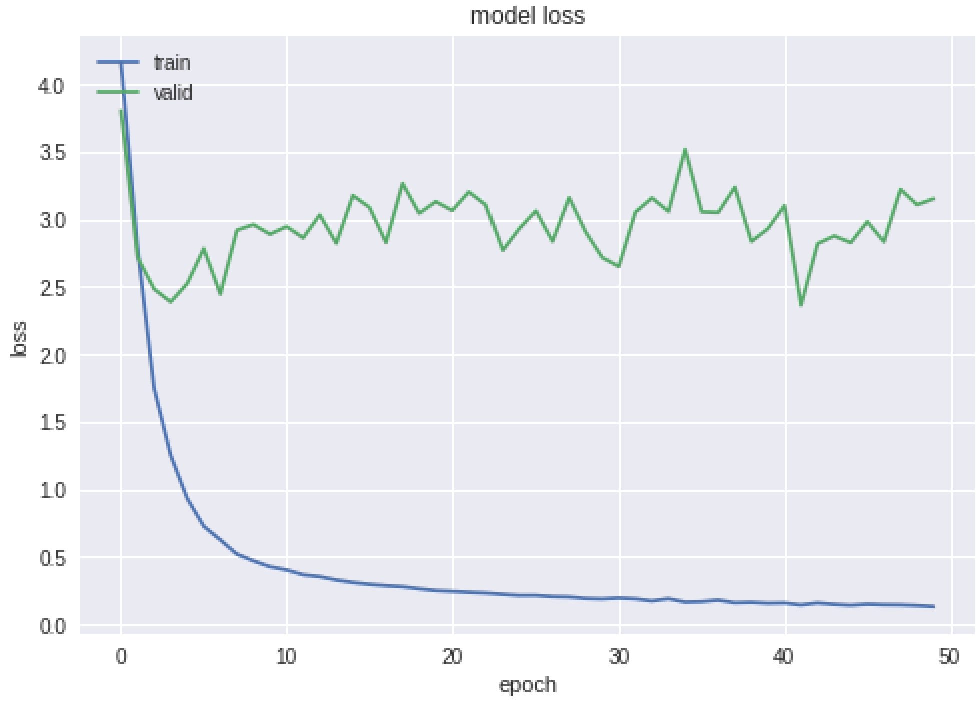
L2 regularization 0.001 + Batch Normalization



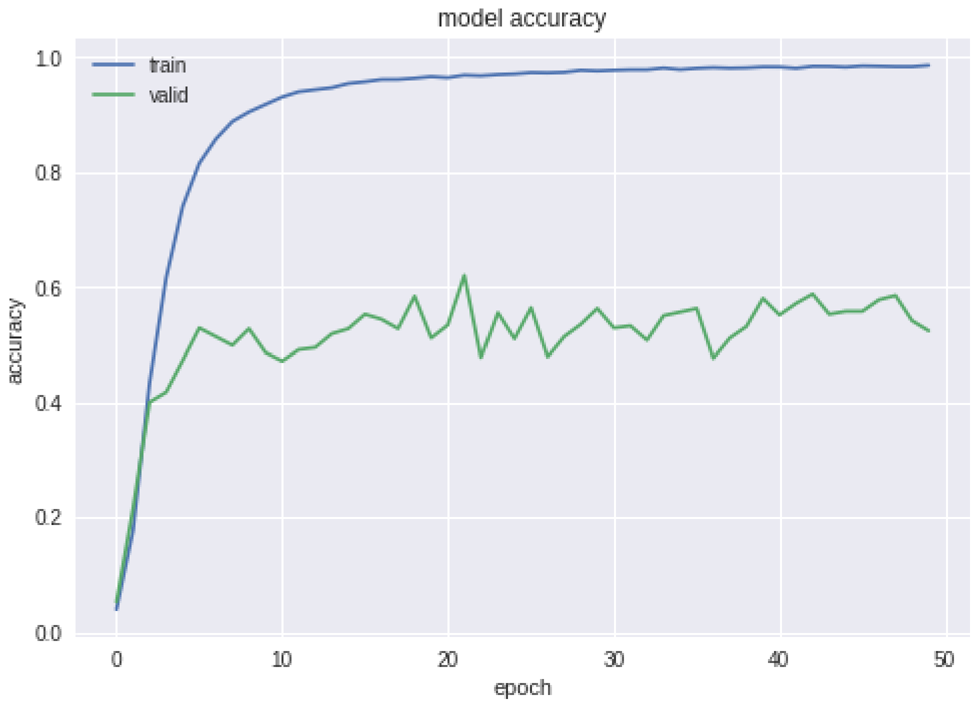


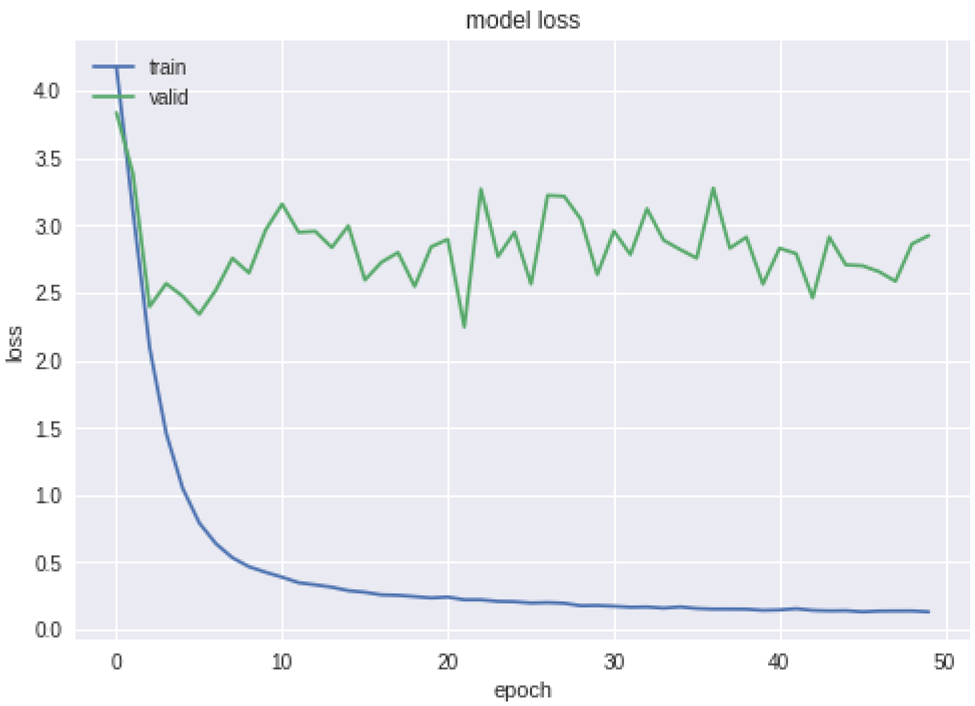
L2 regularization 0.005 + Batch Normalization



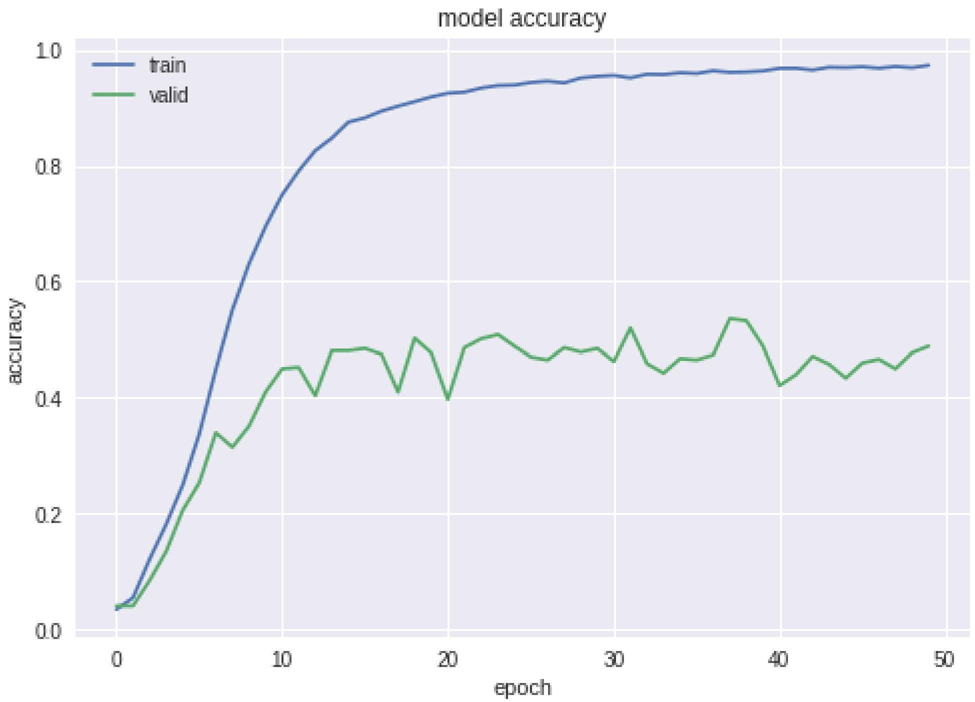


L2 regularization 0.01 + Batch Normalization





L2 regularization 0.1 + Batch Normalization



- **Overfitting**
  - Increase regularization

L2 regularization + Batch Normalization

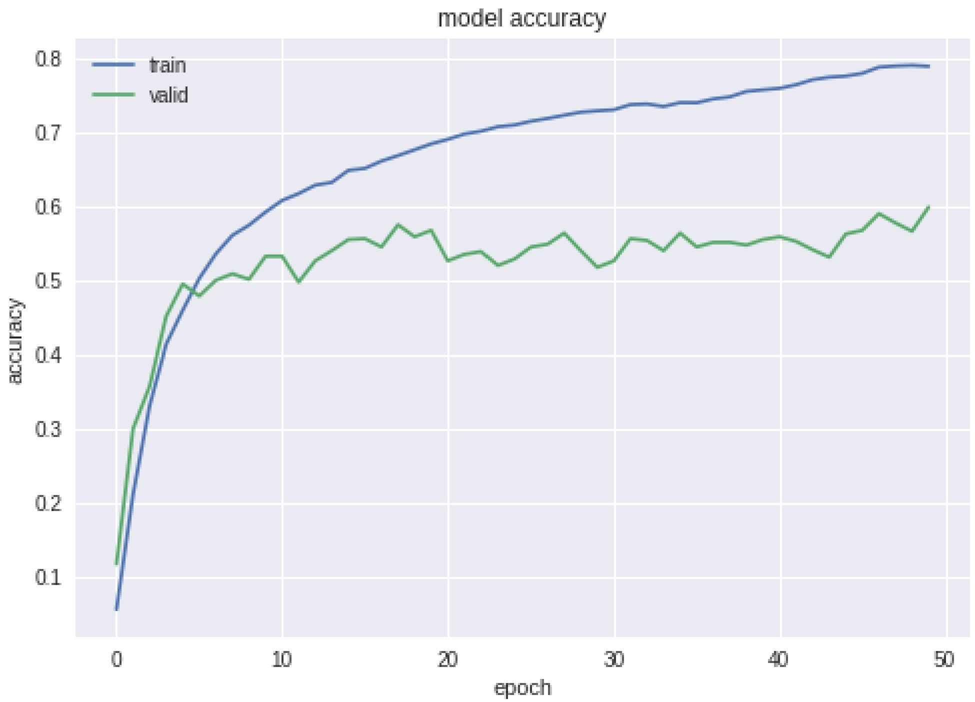
- Comments on using this model
  - Using regularization with batch normalization doesn’t cause an improvement in reducing overfitting in the model.
  - Using regularization allows to use much bigger regularization parameter.
  - After applying batch normalization, the validation accuracy increases with regularization parameter as described in the table.

C	Best validation accuracy
0.001	56%
0.01	**62%**
0.1	53.25%
10	
100	

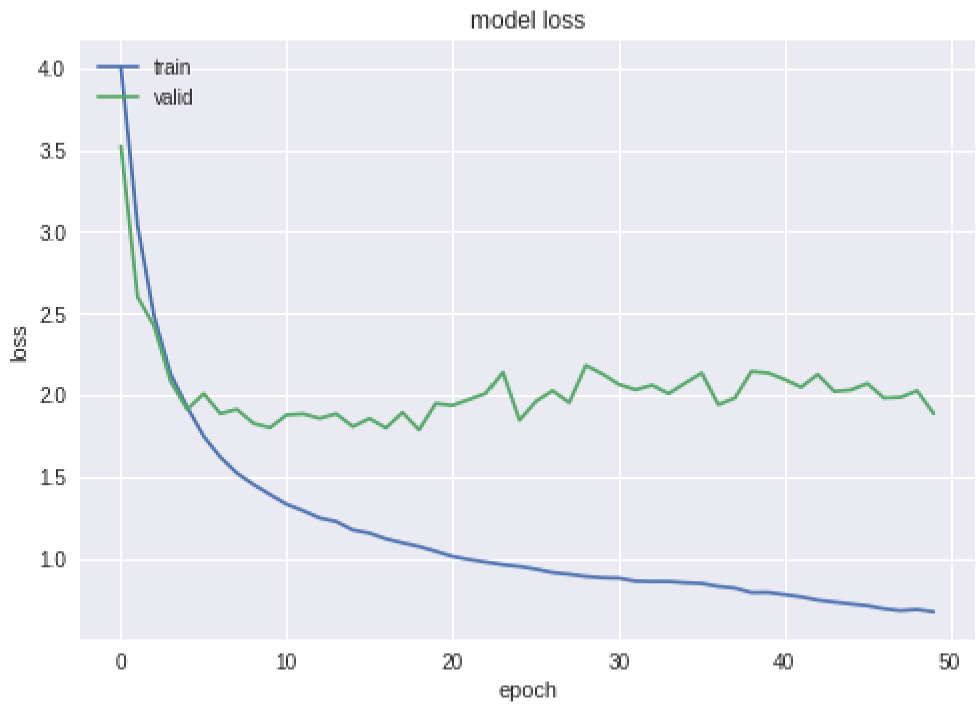
Add dropout to the baseline model

- Use these values for dropout probability
  - 0 (no dropout – baseline model)
  - 0.1
  - 0.3
  - 0.5

Dropout 0.1

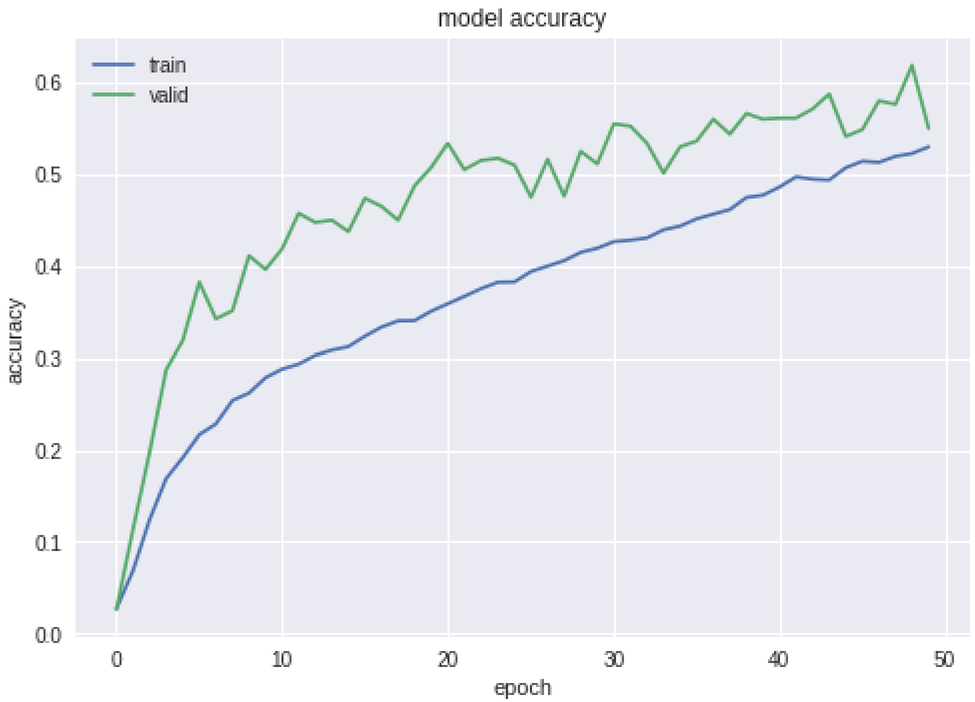


model accuracy
train
valid
accuracy
epoch
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
50



model loss
train
valid
loss
epoch
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0
10
20
30
40
50

Dropout 0.3


model accuracy
train
valid
accuracy
epoch
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30
40
50


model loss
train
valid
loss
epoch
1.5
2.0
2.5
3.0
3.5
4.0
0
10
20
30
40
50

Dropout 0.5

model accuracy
train
valid
accuracy
0.1
0.2
0.3
0.4
0
10
20
30
40
50
epoch

model loss
train
valid
loss
2.50
2.75
3.00
3.25
3.50
3.75
4.00
4.25
0
10
20
30
40
50
epoch

Dropout

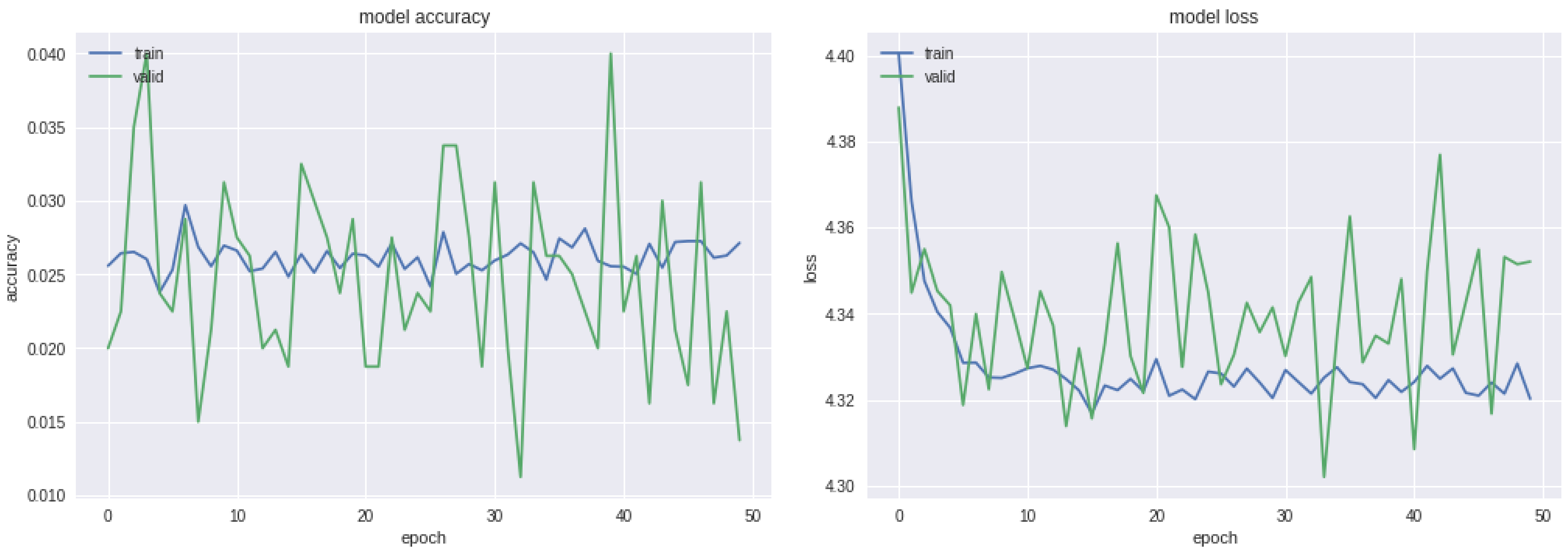
- <u>Comments on using dropout</u>
  - The model with dropout probability 0.3 doesn’t overfit training set.
  - Increasing dropout probability to 0.5 lead to a bad model.

C	**Best validation accuracy**
0.1	59.88%
0.3	**61.88%**
0.5	44.37%

Dropout + Batch Normalization

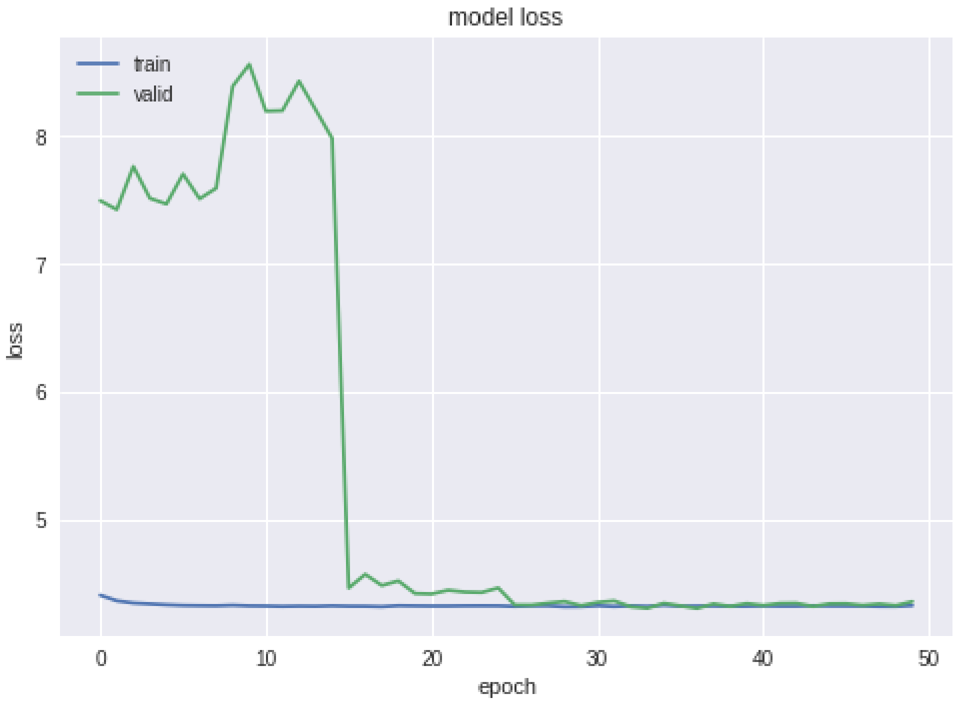
- Apply both dropout along with batch normalization and check the results

Dropout 0.1 + Batch Normalization



Dropout 0.3 + Batch Normalization

Dropout 0.5 + Batch Normalization



Dropout + Batch Normalization

- **It is not clear why applying dropout with batch normalization caused the model to learn nothing.**

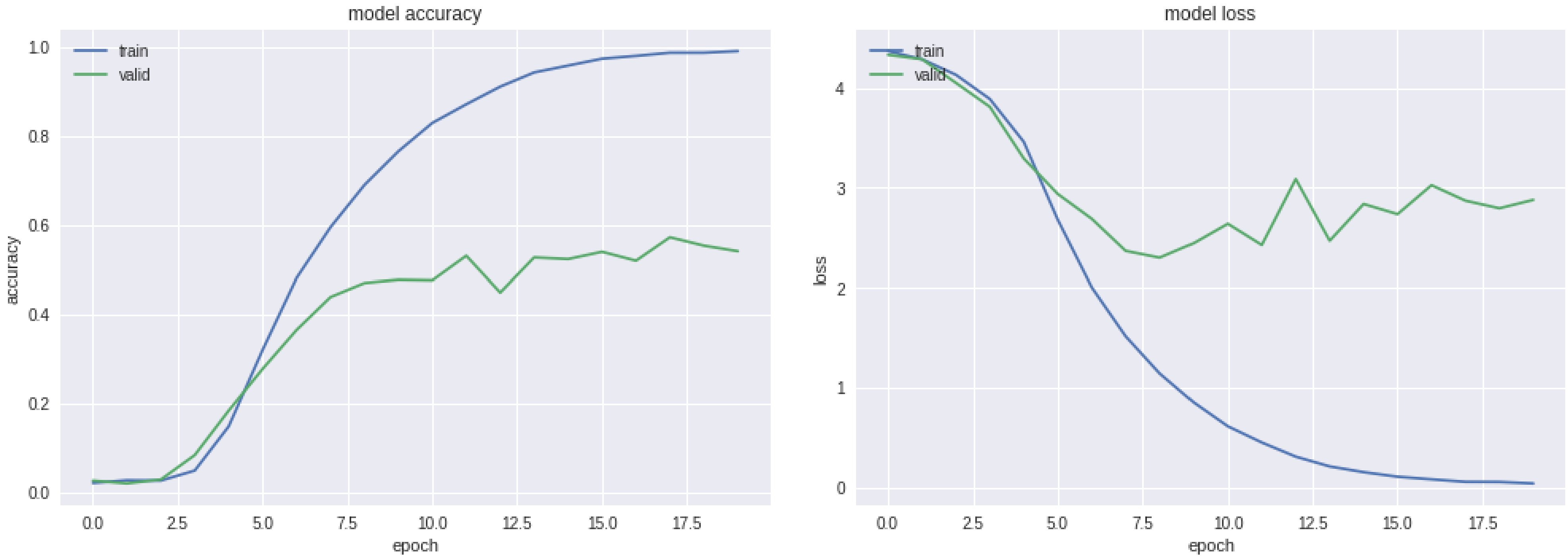
1st Fully Connected Layer Dense(128) instead of Dense(64) Dropout 0.3

- Increasing the number of neurons in Dense layer leads to a better model.
  - Validation accuracy increased from 61.88% to 65.75%.

1st Fully Connected Layer Dense(128) instead of Dense(64) Dropout 0.3 + Batch Normalization

- Increasing the number of neurons in Dense layer leads to a better model.
  - The previous model doesn't learn any thing (so bad).
  - The current model's best validation accuracy is 72.75%.
  - It is the best performance achieved on validation set.
  - The model doesn't seem to be the best model because there is overfitting.

Baseline SGD instead of ADAM



- In general, using ADAM cause the model to converge faster.
- This case is not obvious in this task or model.

- Дополнительные задания (для желающих):
  - Какой dropout лучше использовать для сверточных сетей?
  - В чем разница между softmax loss и center loss? Какой лучше?
  - Что такое архитектура Inception?
  - Как понять, чему обучилась сверточная нейронная сеть? (посмотрите Google DeepDream)

Softmax Loss vs Center Loss

- Softmax Loss
  - Separable, the deep features are not discriminative enough. by intra-class variation

$$\mathcal{L}_S = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}}$$

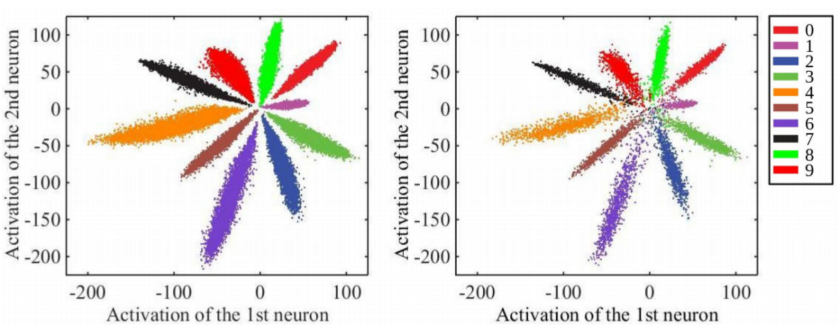
- Center Loss
  - Using Center Loss with softmax leads to a better discriminative model.

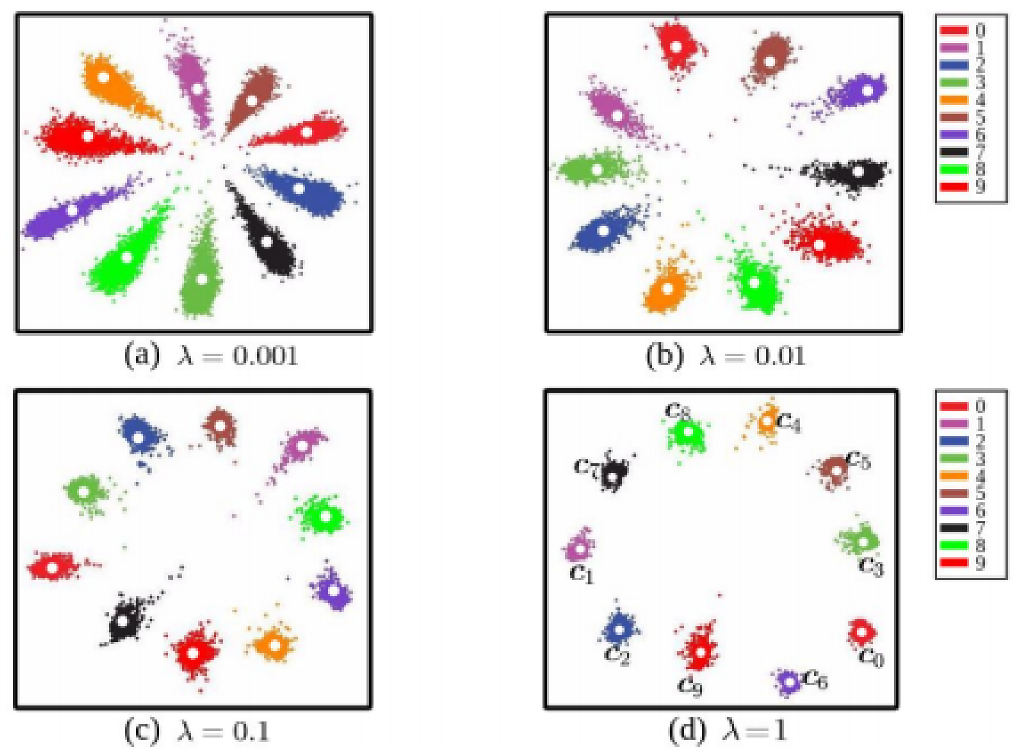
$$\mathcal{L}_C = \frac{1}{2} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C$$

Softmax Loss vs Center Loss

- The affect of using Center Loss is shown in the following figure according to “Wen, Yandong, et al. A Discriminative Feature Learning Approach for Deep Face Recognition”





Inception

- Inception network motivation
  - Szegedy et al. 2014. Going deeper with convolutions
  - You want to apply many types of convolutions or just pooling
    - Convolution, 1x1, 64 filters
    - Convolution, 3x3, 128 filters, same padding
    - Convolution, 5x5, 256 filters, same padding
  - Just apply them all and stack them
    - These filters must have same h,w but different number of channels
    - Using 1by1 convolutions, you can unify the number of channels before concatenation.
    - It is called battle neck layer.
  - There is problem of computational cost

Inception

- Using Inception you can test many architectures at the same time.
- Inception often used with TransferLearning
  - You can use any layer to get high representation of an image (AutoEncoder)
- Inception has many output layers at different depths.

What does Convolution Layer learn?

- The earlier layers learn how to detect simple structures.
- As you go deeper more complex shapes the image can represent
- In style transform, layers in middle are used because they capture style, meanwhile the later layers capture content.

Thank you