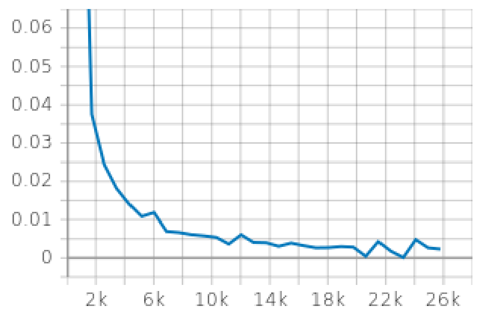
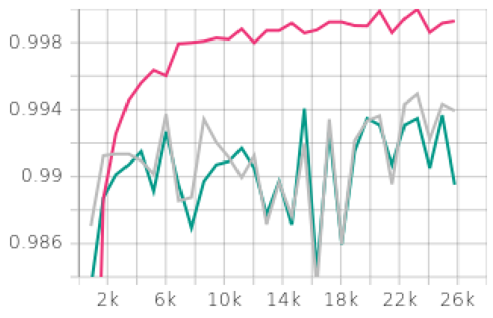
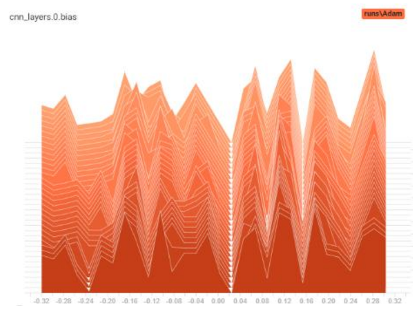
1-1

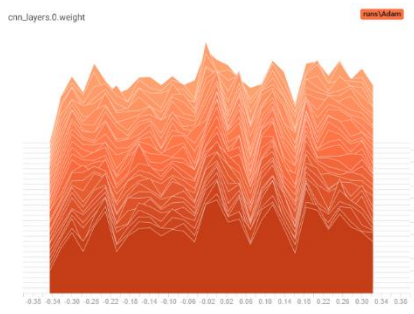
- network architecture：
  input – CNN – pooling – CNN – pooling – CNN – pooling – fully-connect – output
- Large amount of pixels are necessary for the network recognize the object, you may use bigger filters, on other hand if objects are somewhat small or local features, you consider applying smaller filters.
- Small stride size would be better at capturing the details of the input image.
- Learning curve, Accuracy



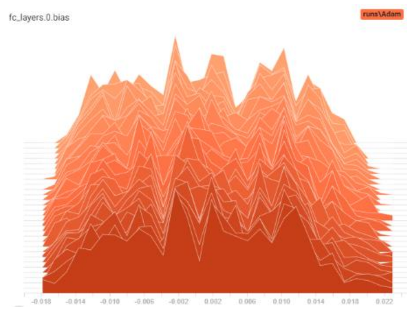


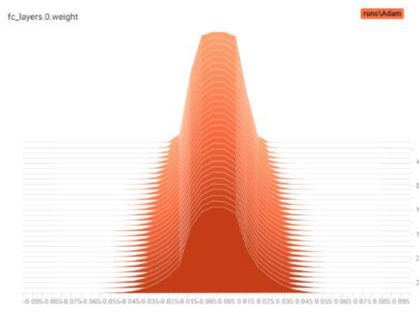
- CNN：bias, weight



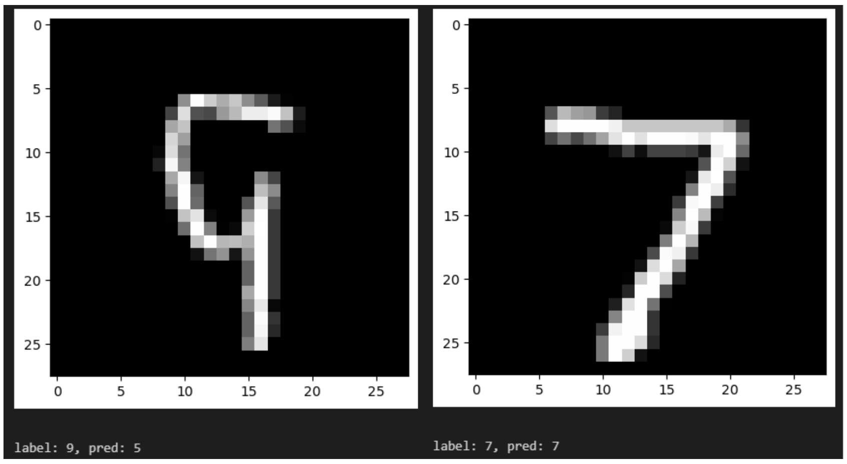


- Fully-connect：bias, weight



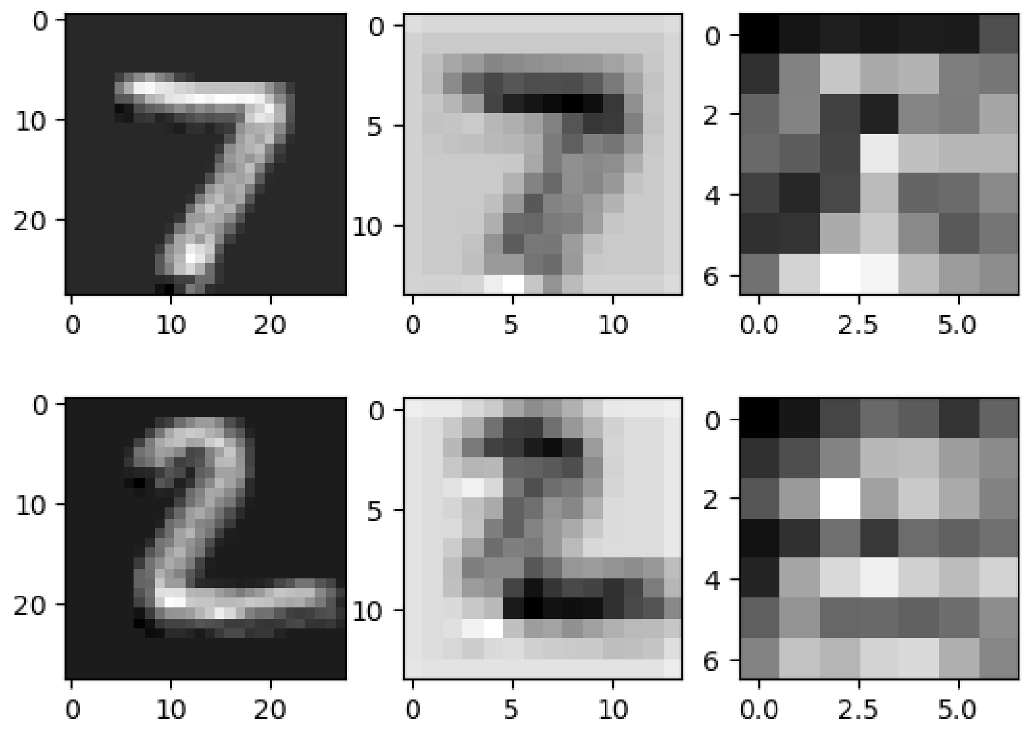


1-2


label: 9, pred: 5

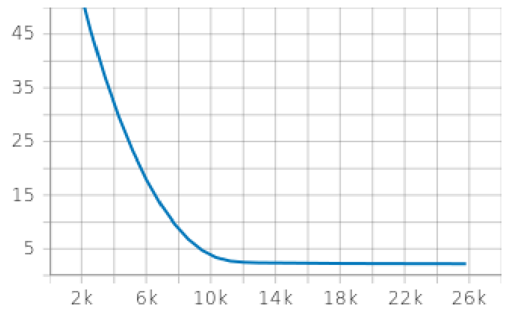
label: 7, pred: 7

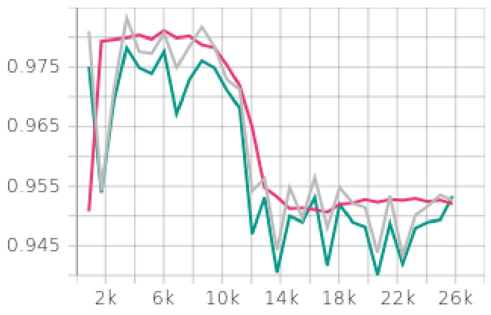
1-3



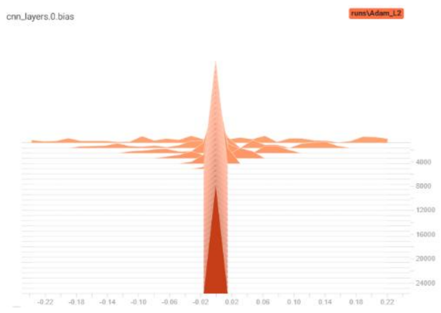
1-4

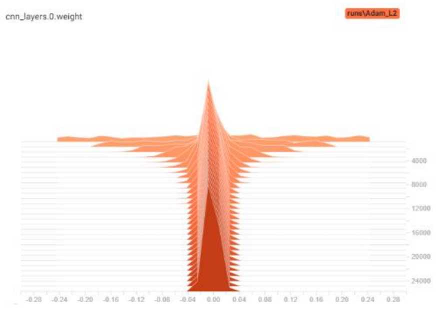
- Learning curve, Accuracy



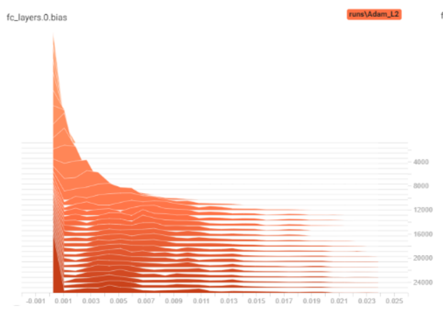


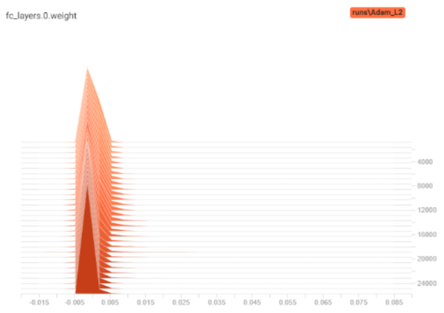
- CNN ： bias, weight





- Fully-connect ： bias, weight

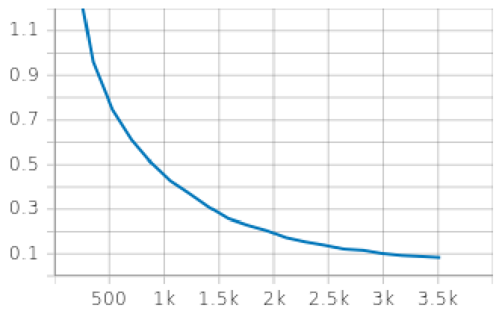


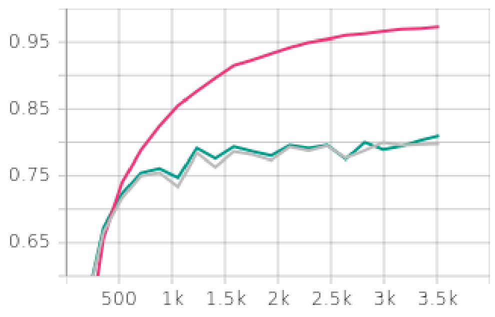


- Learning curve looks smoother, since the L2 regularization. But the loss cannot be lower, due to the learning rate is too small.

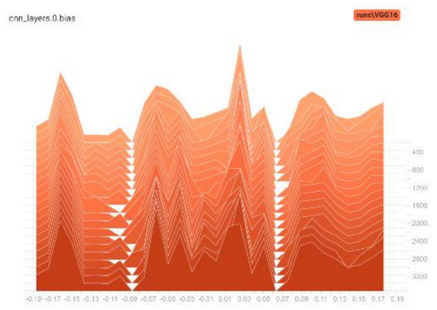
2-1

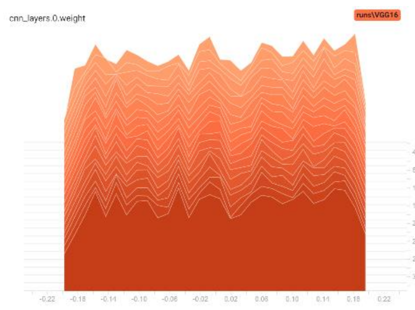
- network architecture：VGG16
- Learning curve, Accuracy



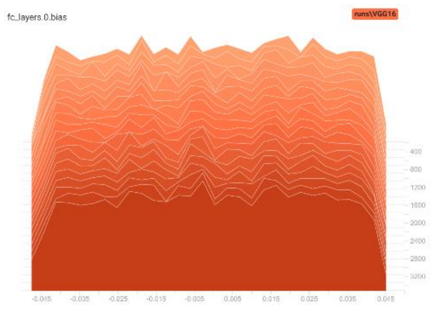


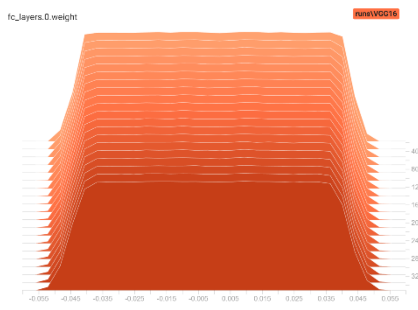
- CNN：bias, weight



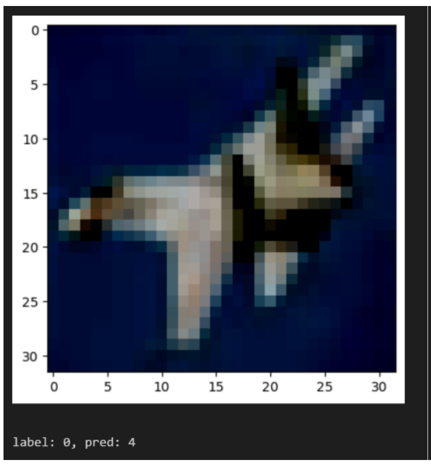


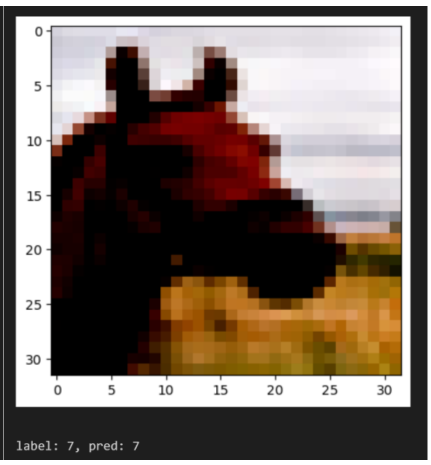
- Fully-connect：bias, weight



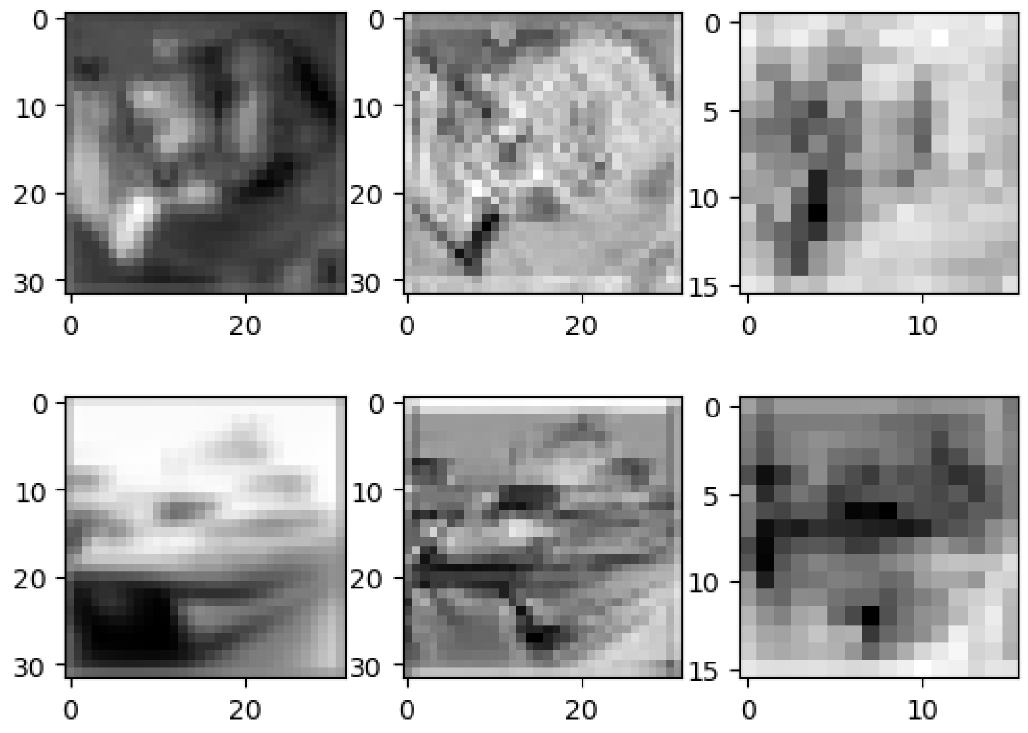


2-2


```
label: 0, pred: 4
```

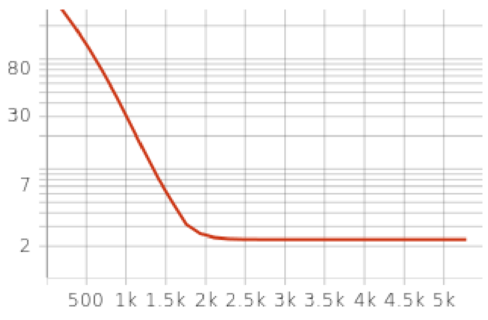

```
label: 7, pred: 7
```

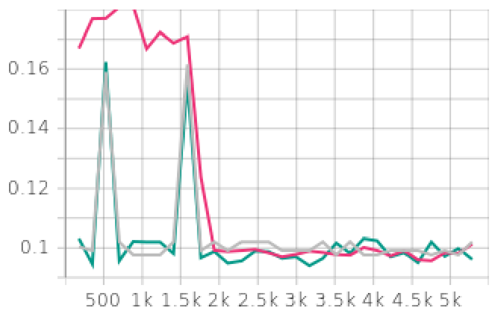
2-3



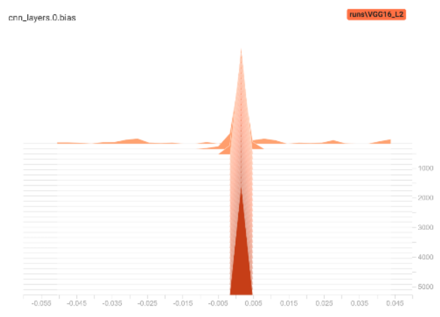
2-4

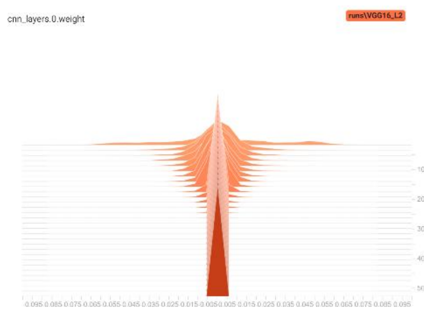
- Learning curve, Accuracy



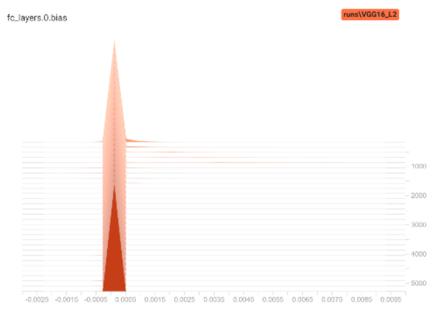


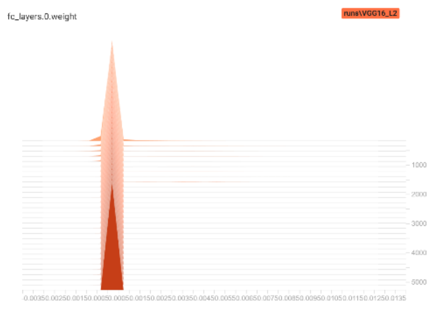
- CNN：bias, weight





- Fully-connect：bias, weight





2-5

- Preprocessing
  transforms.RandomHorizontalFlip()
  transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))