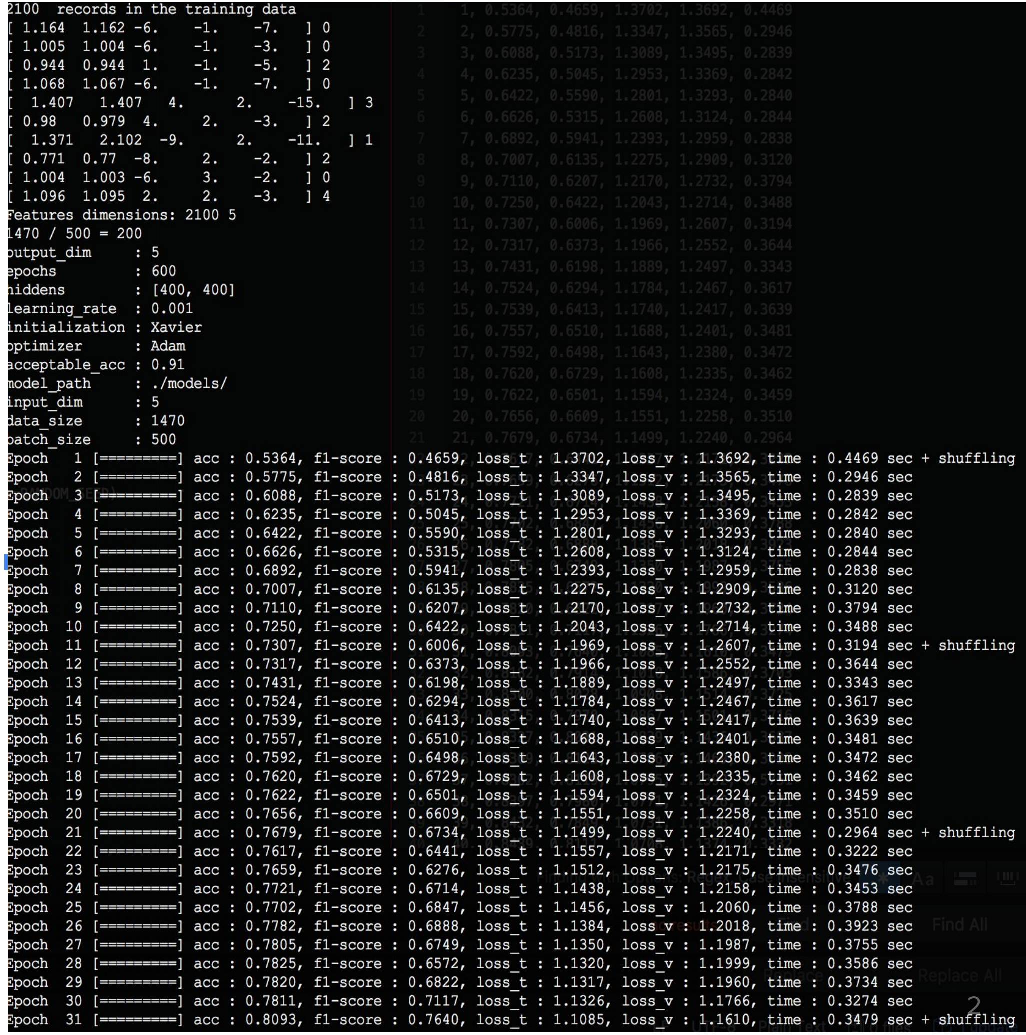
Category_class

Sahli Mohammed
nihon.sahli@gmail.com



```
2100  records in the training data
[ 1.164  1.162 -6.    -1.    -7.   ] 0
[ 1.005  1.004 -6.    -1.    -3.   ] 0
[ 0.944  0.944  1.    -1.    -5.   ] 2
[ 1.068  1.067 -6.    -1.    -7.   ] 0
[  1.407   1.407   4.      2.    -15.   ] 3
[ 0.98   0.979  4.     2.    -3.   ] 2
[  1.371   2.102  -9.      2.    -11.   ] 1
[ 0.771  0.77  -8.     2.    -2.   ] 2
[ 1.004  1.003 -6.     3.    -2.   ] 0
[ 1.096  1.095  2.     2.    -3.   ] 4
Features dimensions: 2100 5
1470 / 500 = 200
output_dim     : 5
epochs         : 600
hiddens        : [400, 400]
learning_rate  : 0.001
initialization : Xavier
optimizer      : Adam
acceptable_acc : 0.91
model_path     : ./models/
input_dim      : 5
data_size      : 1470
batch_size     : 500
Epoch   1 [=========] acc : 0.5364, f1-score : 0.4659, loss_t : 1.3702, loss_v : 1.3692, time : 0.4469 sec + shuffling
Epoch   2 [=========] acc : 0.5775, f1-score : 0.4816, loss_t : 1.3347, loss_v : 1.3565, time : 0.2946 sec
Epoch   3 [=========] acc : 0.6088, f1-score : 0.5173, loss_t : 1.3089, loss_v : 1.3495, time : 0.2839 sec
Epoch   4 [=========] acc : 0.6235, f1-score : 0.5045, loss_t : 1.2953, loss_v : 1.3369, time : 0.2842 sec
Epoch   5 [=========] acc : 0.6422, f1-score : 0.5590, loss_t : 1.2801, loss_v : 1.3293, time : 0.2840 sec
Epoch   6 [=========] acc : 0.6626, f1-score : 0.5315, loss_t : 1.2608, loss_v : 1.3124, time : 0.2844 sec
Epoch   7 [=========] acc : 0.6892, f1-score : 0.5941, loss_t : 1.2393, loss_v : 1.2959, time : 0.2838 sec
Epoch   8 [=========] acc : 0.7007, f1-score : 0.6135, loss_t : 1.2275, loss_v : 1.2909, time : 0.3120 sec
Epoch   9 [=========] acc : 0.7110, f1-score : 0.6207, loss_t : 1.2170, loss_v : 1.2732, time : 0.3794 sec
Epoch  10 [=========] acc : 0.7250, f1-score : 0.6422, loss_t : 1.2043, loss_v : 1.2714, time : 0.3488 sec
Epoch  11 [=========] acc : 0.7307, f1-score : 0.6006, loss_t : 1.1969, loss_v : 1.2607, time : 0.3194 sec + shuffling
Epoch  12 [=========] acc : 0.7317, f1-score : 0.6373, loss_t : 1.1966, loss_v : 1.2552, time : 0.3644 sec
Epoch  13 [=========] acc : 0.7431, f1-score : 0.6198, loss_t : 1.1889, loss_v : 1.2497, time : 0.3343 sec
Epoch  14 [=========] acc : 0.7524, f1-score : 0.6294, loss_t : 1.1784, loss_v : 1.2467, time : 0.3617 sec
Epoch  15 [=========] acc : 0.7539, f1-score : 0.6413, loss_t : 1.1740, loss_v : 1.2417, time : 0.3639 sec
Epoch  16 [=========] acc : 0.7557, f1-score : 0.6510, loss_t : 1.1688, loss_v : 1.2401, time : 0.3481 sec
Epoch  17 [=========] acc : 0.7592, f1-score : 0.6498, loss_t : 1.1643, loss_v : 1.2380, time : 0.3472 sec
Epoch  18 [=========] acc : 0.7620, f1-score : 0.6729, loss_t : 1.1608, loss_v : 1.2335, time : 0.3462 sec
Epoch  19 [=========] acc : 0.7622, f1-score : 0.6501, loss_t : 1.1594, loss_v : 1.2324, time : 0.3459 sec
Epoch  20 [=========] acc : 0.7656, f1-score : 0.6609, loss_t : 1.1551, loss_v : 1.2258, time : 0.3510 sec
Epoch  21 [=========] acc : 0.7679, f1-score : 0.6734, loss_t : 1.1499, loss_v : 1.2240, time : 0.2964 sec + shuffling
Epoch  22 [=========] acc : 0.7617, f1-score : 0.6441, loss_t : 1.1557, loss_v : 1.2171, time : 0.3222 sec
Epoch  23 [=========] acc : 0.7659, f1-score : 0.6276, loss_t : 1.1512, loss_v : 1.2175, time : 0.3476 sec
Epoch  24 [=========] acc : 0.7721, f1-score : 0.6714, loss_t : 1.1438, loss_v : 1.2158, time : 0.3453 sec
Epoch  25 [=========] acc : 0.7702, f1-score : 0.6847, loss_t : 1.1456, loss_v : 1.2060, time : 0.3788 sec
Epoch  26 [=========] acc : 0.7782, f1-score : 0.6888, loss_t : 1.1384, loss_v : 1.2018, time : 0.3923 sec
Epoch  27 [=========] acc : 0.7805, f1-score : 0.6749, loss_t : 1.1350, loss_v : 1.1987, time : 0.3755 sec
Epoch  28 [=========] acc : 0.7825, f1-score : 0.6572, loss_t : 1.1320, loss_v : 1.1999, time : 0.3586 sec
Epoch  29 [=========] acc : 0.7820, f1-score : 0.6822, loss_t : 1.1317, loss_v : 1.1960, time : 0.3734 sec
Epoch  30 [=========] acc : 0.7811, f1-score : 0.7117, loss_t : 1.1326, loss_v : 1.1766, time : 0.3274 sec
Epoch  31 [=========] acc : 0.8093, f1-score : 0.7640, loss_t : 1.1085, loss_v : 1.1610, time : 0.3479 sec + shuffling
```

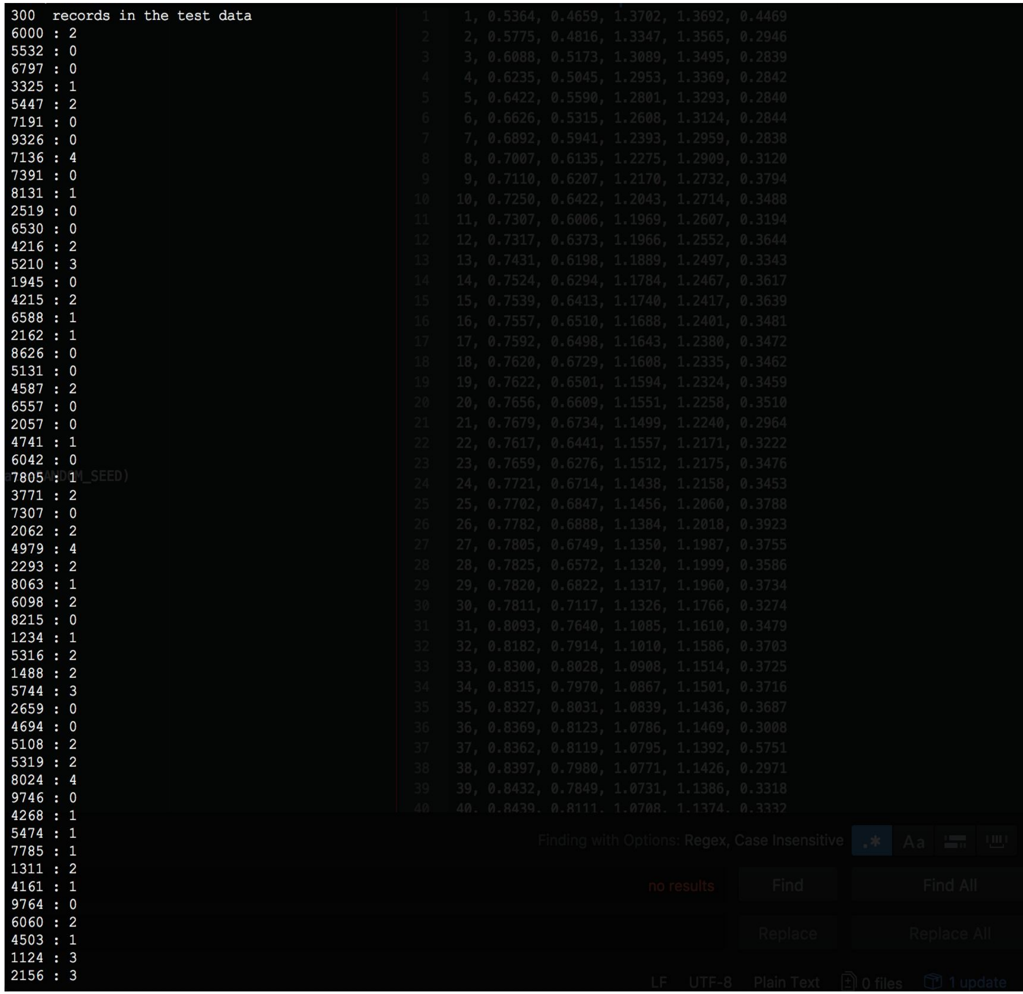
Manual

How to train:

```
$ python3 train.py train "data/train.csv"
```

How to predict:

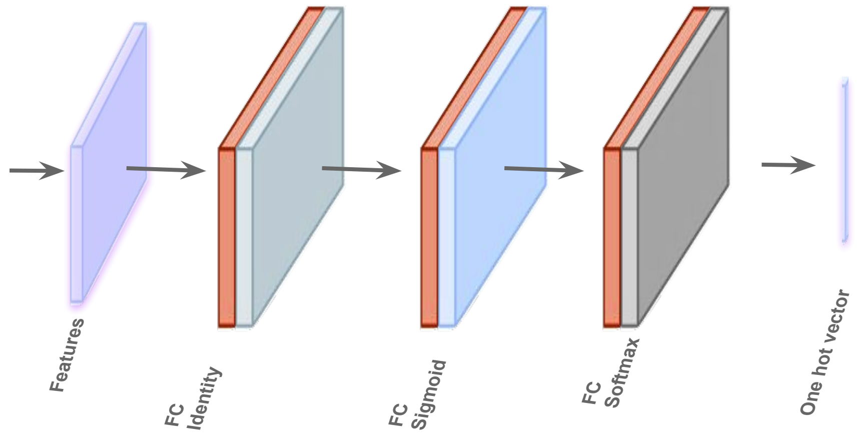
```
$ python3 train.py predict "data/test.csv" models_/541.ckpt
```



```
300  records in the test data
6000 : 2
5532 : 0
6797 : 0
3325 : 1
5447 : 2
7191 : 0
9326 : 0
7136 : 4
7391 : 0
8131 : 1
2519 : 0
6530 : 0
4216 : 2
5210 : 3
1945 : 0
4215 : 2
6588 : 1
2162 : 1
8626 : 0
5131 : 0
4587 : 2
6557 : 0
2057 : 0
4741 : 1
6042 : 0
7805 : 1
3771 : 2
7307 : 0
2062 : 2
4979 : 4
2293 : 2
8063 : 1
6098 : 2
8215 : 0
1234 : 1
5316 : 2
1488 : 2
5744 : 3
2659 : 0
4694 : 0
5108 : 2
5319 : 2
8024 : 4
9746 : 0
4268 : 1
5474 : 1
7785 : 1
1311 : 2
4161 : 1
9764 : 0
6060 : 2
4503 : 1
1124 : 3
2156 : 3
```

```
 1, 0.5364, 0.4659, 1.3702, 1.3692, 0.4469
 2, 0.5775, 0.4816, 1.3347, 1.3565, 0.2946
 3, 0.6088, 0.5173, 1.3089, 1.3495, 0.2839
 4, 0.6235, 0.5045, 1.2953, 1.3369, 0.2842
 5, 0.6422, 0.5590, 1.2801, 1.3293, 0.2840
 6, 0.6626, 0.5315, 1.2608, 1.3124, 0.2844
 7, 0.6892, 0.5941, 1.2393, 1.2959, 0.2838
 8, 0.7007, 0.6135, 1.2275, 1.2909, 0.3120
 9, 0.7110, 0.6207, 1.2170, 1.2732, 0.3794
10, 0.7250, 0.6422, 1.2043, 1.2714, 0.3488
11, 0.7307, 0.6006, 1.1969, 1.2607, 0.3194
12, 0.7317, 0.6373, 1.1966, 1.2552, 0.3644
13, 0.7431, 0.6198, 1.1889, 1.2497, 0.3343
14, 0.7524, 0.6294, 1.1784, 1.2467, 0.3617
15, 0.7539, 0.6413, 1.1740, 1.2417, 0.3639
16, 0.7557, 0.6510, 1.1688, 1.2401, 0.3481
17, 0.7592, 0.6498, 1.1643, 1.2380, 0.3472
18, 0.7620, 0.6729, 1.1608, 1.2335, 0.3462
19, 0.7622, 0.6501, 1.1594, 1.2324, 0.3459
20, 0.7656, 0.6609, 1.1551, 1.2258, 0.3510
21, 0.7679, 0.6734, 1.1499, 1.2240, 0.2964
22, 0.7617, 0.6441, 1.1557, 1.2171, 0.3222
23, 0.7659, 0.6276, 1.1512, 1.2175, 0.3476
24, 0.7721, 0.6714, 1.1438, 1.2158, 0.3453
25, 0.7702, 0.6847, 1.1456, 1.2060, 0.3788
26, 0.7782, 0.6888, 1.1384, 1.2018, 0.3923
27, 0.7805, 0.6749, 1.1350, 1.1987, 0.3755
28, 0.7825, 0.6572, 1.1320, 1.1999, 0.3586
29, 0.7820, 0.6822, 1.1317, 1.1960, 0.3734
30, 0.7811, 0.7117, 1.1326, 1.1766, 0.3274
31, 0.8093, 0.7640, 1.1085, 1.1610, 0.3479
32, 0.8182, 0.7914, 1.1010, 1.1586, 0.3703
33, 0.8300, 0.8028, 1.0908, 1.1514, 0.3725
34, 0.8315, 0.7970, 1.0867, 1.1501, 0.3716
35, 0.8327, 0.8031, 1.0839, 1.1436, 0.3687
36, 0.8369, 0.8123, 1.0786, 1.1469, 0.3008
37, 0.8362, 0.8119, 1.0795, 1.1392, 0.5751
38, 0.8397, 0.7980, 1.0771, 1.1426, 0.2971
39, 0.8432, 0.7849, 1.0731, 1.1386, 0.3318
40, 0.8439, 0.8111, 1.0708, 1.1374, 0.3332
```

Architecture



Explanation

Features extraction & Engineering:

The chosen features are : Sold Price, Price, Area Name, Condition, Size

Condition and Area name were simply converted into numbers as follows:

```
condition = {"Fair": -1, "Good": 2, "Like New": 3}
area = {"aaa": 1,"bbb": 2,"ccc": 3,"ddd": 4,"eee": 5,"fff": -6,"ggg": -7,"hhh": -8,"jjj": -9,"kkk": -10}
units = line.split(",")
X = [float(units[2]) / 1000.0,float(units[3]) / 1000.0, float(area[units[4]]), float(condition[units[5]]), -float(units[6])]
```

Dataset is so small (~ 700 points), the first implementation suffered from overfitting,
I solved the problem by augmenting the data 3 times.

Model training:

Weights were initialized by Xavier Initialization
Cross Entropy with logit was chosen as a loss function
Adam optimizer with a batch size of 500 was used for training.
Data were shuffled every 10 epochs.

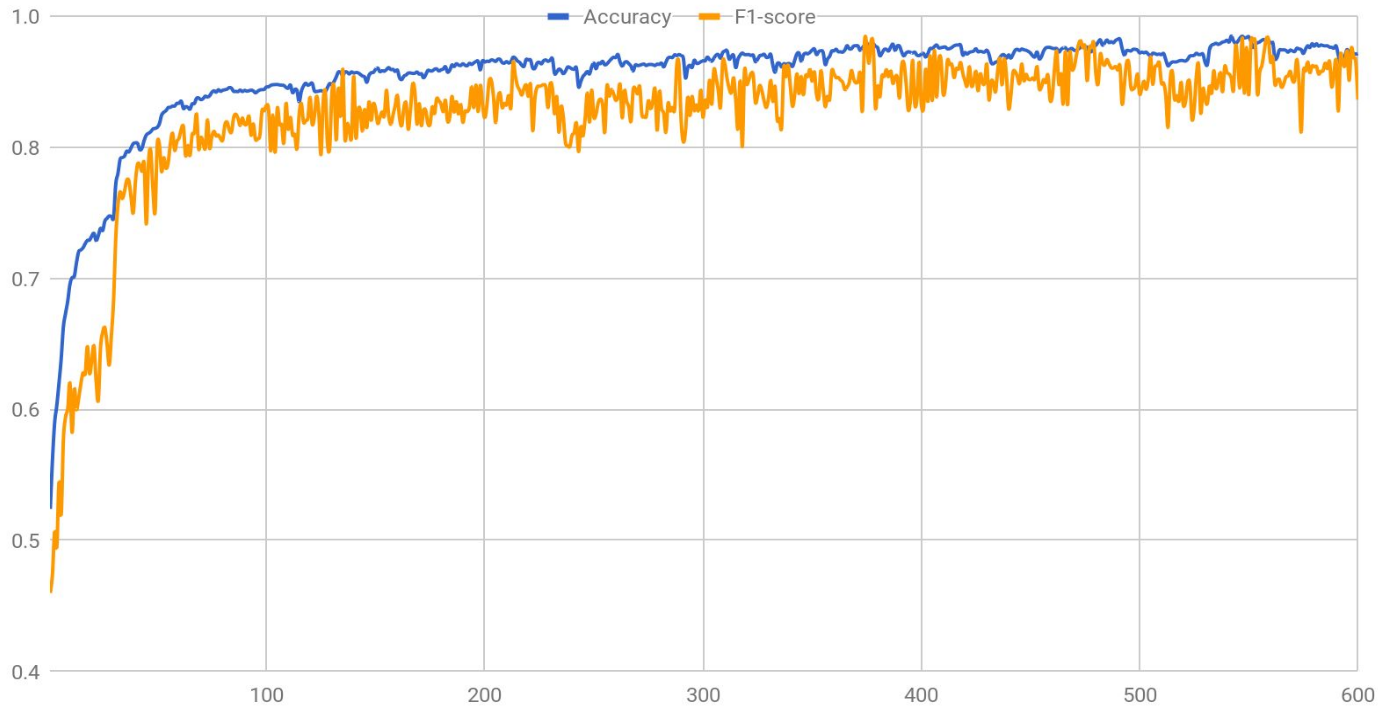
Model Evaluation:

The augmented dataset was divided into 70% for training, and 30% for validation.
The accuracy and losses are shown below.

Evaluation

Epochs	600
Samples	700
Augmentation	3X
Accuracy	0.9336
F1-score	0.8981
Loss_train	0.9704
Loss_valid	1.0398
Time (sec)	18

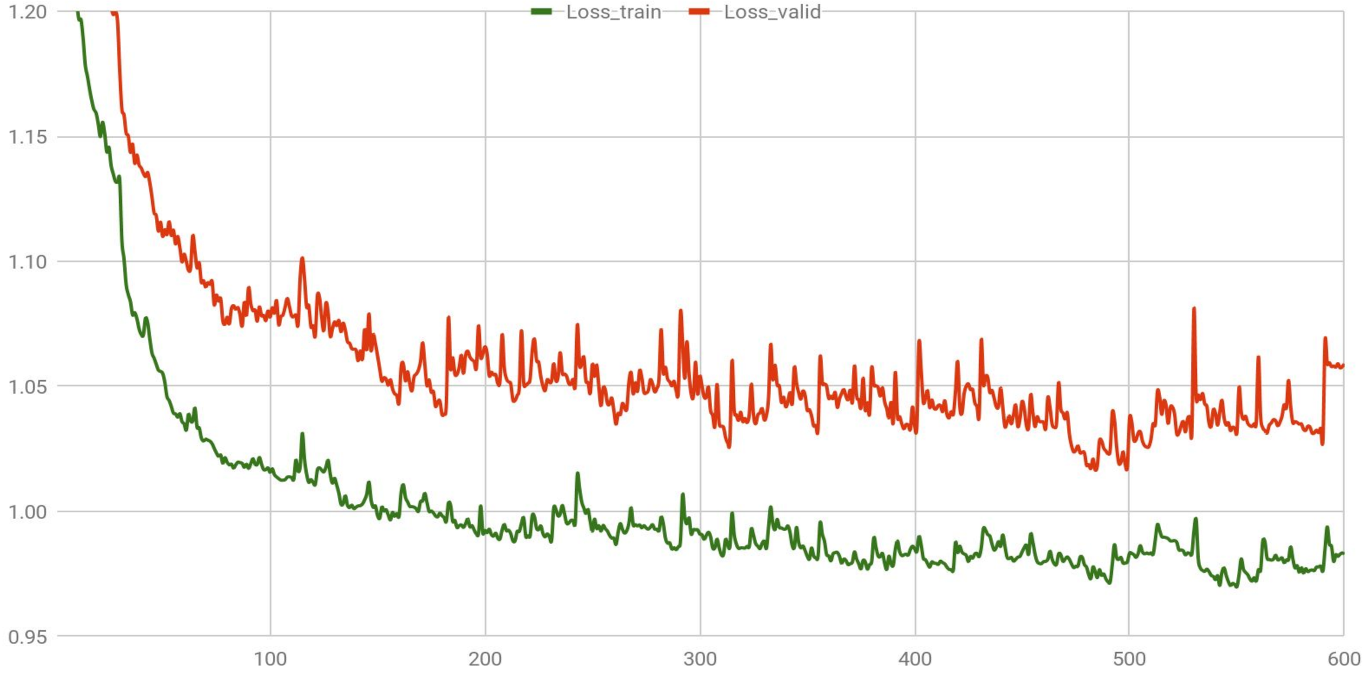
Steps	10
Hidden 1	400
Hidden 2	400
learning rate	0.001
Batch size	500
Initialization	Xavier
Optimizer	Adam
Loss Function	Softmax Cross Entropy
Framework	Tensorflow 1.3
Python	3



Evaluation

Epochs	600
Samples	700
Augmentation	3X
Accuracy	0.9336
F1-score	0.8981
Loss_train	0.9704
Loss_valid	1.0398
Time (sec)	18

Steps	10
Hidden 1	400
Hidden 2	400
learning rate	0.001
Batch size	500
Initialization	Xavier
Optimizer	Adam
Loss Function	Softmax Cross Entropy
Framework	Tensorflow 1.3
Python	3



Prediction

6000 : 2	2293 : 2	3196 : 2	8451 : 3	1712 : 0	7278 : 2	3035 : 2	5553 : 2	9897 : 0	2654 : 0
5532 : 0	8063 : 1	3012 : 4	3495 : 2	3604 : 4	3177 : 2	7108 : 2	9574 : 0	3285 : 0	9126 : 2
6797 : 0	6098 : 2	1824 : 0	9239 : 1	3266 : 2	7103 : 0	9304 : 3	5056 : 0	5428 : 0	1956 : 0
3325 : 1	8215 : 0	4092 : 1	7576 : 0	7358 : 0	2534 : 3	3649 : 2	9740 : 2	7093 : 2	1753 : 0
5447 : 2	1234 : 1	1545 : 1	1752 : 1	2256 : 4	3056 : 1	7183 : 1	1452 : 0	6897 : 2	7620 : 2
7191 : 0	5316 : 2	3360 : 0	7024 : 2	3100 : 0	7883 : 3	4043 : 0	1245 : 0	2457 : 3	8937 : 4
9326 : 0	1488 : 2	3018 : 1	4805 : 2	1611 : 2	3983 : 1	4083 : 1	6169 : 0	1927 : 3	4040 : 1
7136 : 4	5744 : 3	9263 : 0	9218 : 0	8159 : 2	3449 : 0	2289 : 2	5370 : 2	5199 : 0	9576 : 2
7391 : 0	2659 : 0	7755 : 4	3090 : 2	2704 : 0	4019 : 0	8600 : 2	8817 : 4	9705 : 2	9085 : 2
8131 : 1	4694 : 0	7660 : 0	7565 : 0	6684 : 0	9095 : 0	3092 : 0	7159 : 3	4031 : 1	5454 : 1
2519 : 0	5108 : 2	9818 : 2	8397 : 1	4733 : 0	4061 : 0	5897 : 3	1596 : 1	3383 : 1	2465 : 0
6530 : 0	5319 : 2	2052 : 2	5726 : 4	9712 : 2	9243 : 3	8923 : 1	1899 : 3	3001 : 4	5053 : 1
4216 : 2	8024 : 4	2960 : 1	1412 : 1	6304 : 1	2916 : 2	5467 : 1	9259 : 3	8087 : 2	5415 : 1
5210 : 3	9746 : 0	9212 : 3	2594 : 1	4655 : 3	2976 : 2	5218 : 0	5967 : 2	5718 : 2	5260 : 2
1945 : 0	4268 : 1	5646 : 0	4252 : 0	1386 : 1	3465 : 0	6843 : 0	3076 : 4	6959 : 3	8254 : 2
4215 : 2	5474 : 1	3975 : 0	8096 : 0	2241 : 2	4985 : 0	9976 : 2	9373 : 2	1094 : 2	3091 : 1
6588 : 1	7785 : 1	9511 : 3	7002 : 2	2448 : 2	3770 : 2	5594 : 2	8610 : 3	2880 : 2	5554 : 4
2162 : 1	1311 : 2	5408 : 0	2592 : 1	2832 : 1	4884 : 1	6484 : 2	4447 : 0	2403 : 0	2203 : 2
8626 : 0	4161 : 1	1685 : 2	4993 : 3	8188 : 0	9497 : 2	9531 : 4	9285 : 2	6048 : 3	3229 : 2
5131 : 0	9764 : 0	3130 : 4	3877 : 1	2118 : 0	3458 : 2	2393 : 2	5958 : 1	9135 : 0	9597 : 1
4587 : 2	6060 : 2	8437 : 0	6894 : 0	1185 : 1	9944 : 0	2909 : 2	1325 : 1	3750 : 1	7762 : 0
6557 : 0	4503 : 1	8896 : 0	7772 : 0	6161 : 4	4405 : 2	9378 : 2	2386 : 2	9842 : 1	7311 : 2
2057 : 0	1124 : 3	8231 : 1	2454 : 3	6555 : 1	6585 : 0	4133 : 1	2933 : 2	9279 : 2	7643 : 3
4741 : 1	2156 : 3	9032 : 0	7101 : 0	6265 : 4	1075 : 0	7401 : 2	9069 : 1	3138 : 2	3973 : 0
6042 : 0	4442 : 2	1006 : 0	7792 : 2	9980 : 0	7925 : 1	9240 : 2	9280 : 1	3620 : 0	9931 : 1
7805 : 1	8839 : 2	3218 : 2	7492 : 2	5357 : 0	4391 : 1	5351 : 2	4898 : 0	6478 : 0	4278 : 0
3771 : 2	3224 : 2	2826 : 2	6996 : 3	7171 : 3	3396 : 2	5678 : 4	7328 : 1	5667 : 2	9981 : 3
7307 : 0	6085 : 1	8999 : 1	2082 : 0	9855 : 1	4108 : 2	4308 : 3	9720 : 3	7409 : 2	1450 : 1
2062 : 2	7373 : 0	2523 : 1	2928 : 0	1208 : 4	8703 : 1	7592 : 2	7859 : 0	6567 : 1	6929 : 0
4979 : 4	6349 : 1	7835 : 1	8598 : 2	6124 : 2	3013 : 1	1843 : 2	1679 : 0	2088 : 1	1952 : 1