Machine Learning (2016 Fall) HW3 Report

Semi-supervised Learning in Picture Classification

Student : 沈怡廷 r05943001

Kaggle_ID : r05943001_Panda

A. Supervised learning

a. Layer structure

```
model = Sequential()

model.add(Convolution2D(32, 3, 3, border_mode='same',
                        input_shape=X_train_label.shape[1:]))
model.add(Activation('relu'))
model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Convolution2D(64, 3, 3, border_mode='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))
```

b. Performance

acc_train	acc_valid	acc_public
0.8807	0.7260	0.7090

c. Description

Batch size	32	Optimizer	Adam
Epoch	200	Training set	4500
Data augment	Apply	Validation set	500

B. Semi-supervised learning (1) – self-learning

a. Predict and add unlabel data to training set if confident enough

```
if self_learning:
    unlabel_predict_prob  = model.predict_proba(X_unlabel)
    unlabel_predict_class = model.predict_classes(X_unlabel)

    X_train_add = X_unlabel[np.amax(unlabel_predict_prob, axis=1) >= threshold]
    X_train_new = np.concatenate((X_train_label, X_train_add), axis=0)

    Y_train_temp = np.array([], dtype=np.int64)
    Y_train_temp = np.append(Y_train_temp, unlabel_predict_class[np.amax(unlabel_predict_prob, axis=1) >= threshold], axis=0)
    Y_train_temp = np.resize(Y_train_temp, (Y_train_temp.shape[0], 1))
    Y_train_add  = np_utils.to_categorical(Y_train_temp, nb_classes)
    Y_train_new  = np.concatenate((Y_train_label, Y_train_add), axis=0)
```

b. Performance

(a) Without using testing data

acc_train	acc_valid	acc_public
0.9024	0.7440	0.7150

(b) Use testing data meanwhile

acc_train	acc_valid	acc_public
0.9120	0.7240	0.7174

c. Description

Batch size	32	Optimizer	Adam
Epoch	200	Training set	4500
Data augment	Apply	Validation set	500
Threshold	0.9	Iteration	2

C. Semi-supervised learning (2) – clustering

a. Train an autoencoder to extract feature

```
input_img = Input(shape=(3, 32, 32))

x = Convolution2D(8, 3, 3, activation='relu', border_mode='same')(input_img)
x = MaxPooling2D((2, 2), border_mode='same')(x)
x = Convolution2D(16, 3, 3, activation='relu', border_mode='same')(x)
x = MaxPooling2D((2, 2), border_mode='same')(x)
x = Convolution2D(32, 3, 3, activation='relu', border_mode='same')(x)
x = MaxPooling2D((2, 2), border_mode='same')(x)
x = Convolution2D(64, 3, 3, activation='relu', border_mode='same')(x)
x = MaxPooling2D((2, 2), border_mode='same')(x)
x = Convolution2D(128, 3, 3, activation='relu', border_mode='same')(x)
x = MaxPooling2D((2, 2), border_mode='same')(x)
encoded = Convolution2D(256, 3, 3, activation='relu', border_mode='same')(x)

# at this point the representation is (256, 1, 1) i.e. 256-dimensional

x = Convolution2D(128, 3, 3, activation='relu', border_mode='same')(encoded)
x = UpSampling2D((2, 2))(x)
x = Convolution2D(64, 3, 3, activation='relu', border_mode='same')(x)
x = UpSampling2D((2, 2))(x)
x = Convolution2D(32, 3, 3, activation='relu', border_mode='same')(x)
x = UpSampling2D((2, 2))(x)
x = Convolution2D(16, 3, 3, activation='relu', border_mode='same')(x)
x = UpSampling2D((2, 2))(x)
x = Convolution2D(8, 3, 3, activation='relu', border_mode='same')(x)
x = UpSampling2D((2, 2))(x)
decoded = Convolution2D(3, 3, 3, activation='sigmoid', border_mode='same')(x)
```

b. Clustering by cosine similarity

```
X_label_code   = encoder.predict(X_label)
X_unlabel_code = encoder.predict(X_unlabel)

X_label_code   = np.reshape(X_label_code, (5000, 256))
X_unlabel_code = np.reshape(X_unlabel_code, (45000, 256))

cos_sim = cosine_similarity(X_unlabel_code, X_label_code)
cos_sim = np.reshape(cos_sim, (45000, 10, 500))
cos_sim = np.average(cos_sim, axis=2)
y_unlabel = np.argmax(cos_sim, axis=1)
```

c. Performance

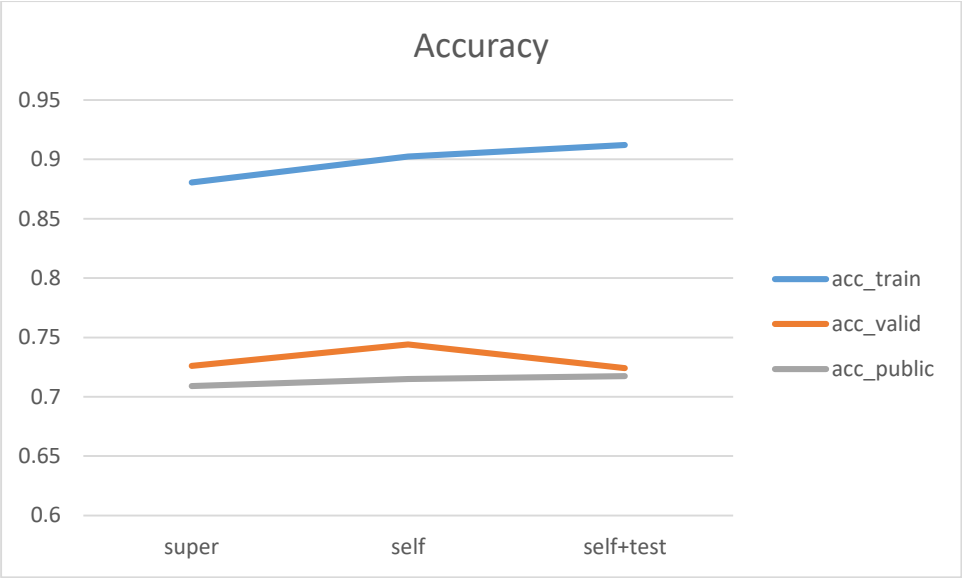
acc_train	acc_valid	acc_public
0.8175	0.2480	0.2322

d. Description

Batch size	32	Optimizer	Adam
Epoch	200	Training set	4500
Data augment	Apply	Validation set	500

D. Compare and analysis

a. Accuracy



Note : 使用 self-learning 可以有效提升 accuracy

Note : 將 testing data 放入，若預測不準確，可能會讓 testing data 上面的表現變差

b. Bad performance of auto-encoder

方法(2)反而會導致效果變差，可能的原因是，剛開始利用 auto-encoder 分群時沒有做好，導致 training 時使用到錯誤的 label。