**Bài thực hành về nhà**

*Nguyễn Hồng Đăng*

**Nội dung:**

1. Ở bài tập về nhà số 11, bạn đã làm quen với bài toán phân loại cảm xúc trên bộ dữ liệu FER-2013 sử dụng phương pháp trích xuất đặc trưng BoW kết hợp với mô hình phân loại SVM, Random forest, KNN. Trong bài tập này, bạn hãy thử sử dụng mạng LeNet và BKNet ở trên để huấn luyện mô hình nhận diện cảm xúc.

2. Ở bài tập 1 bạn đã sử dụng các mô hình có sẵn để huấn luyện mô hình nhận diện cảm xúc. Ở bài tập này, hãy thử lập trình mô hình đề xuất trong bài báo dưới đây cho bài toán nhận diện cảm xúc: <https://www.researchgate.net/profile/Dinh_Sang/publication/321257241_Facial_expression_recognition_using_deep_convolutional_neural_networks/links/5b12a7824585150a0a619d6c/Facial-expression-recognition-using-deep-convolutional-neural-networks.pdf>

3. Tìm hiểu kỹ thuật sử dụng các mô hình mạng CNN kinh điển như VGG, ResNet, DenseNet với framework Keras, thử áp dụng cho các bài toán nhận diện cảm xúc, phân loại phương tiện giao thông: <https://keras.io/applications/>

**1. Sử dụng mạng LeNet và BKNet ở trên để huấn luyện mô hình nhận diện cảm xúc.**

Thử nghiệm với 2 mạng LeNet và BKNet, không sử dụng thêm data augmentation, kết quả cho thấy BKNet có độ chính xác tốt hơn LeNet.

**1.1. LeNet**

Sử dụng 2 callbacks:

+ Model checkpoint lưu lại model với accuracy trên tập validation (chọn là tập public test) lớn nhất.

+ ReduceLROnPlateau để giảm learning rate đi 10 lần khi loss không giảm trong 10 epochs

+ Sử dụng adam optimizer, cross entropy loss

**Kết quả:**

- Accuracy on public test set: 0.5185288381247708

- Accuracy on private test set: 0.4993034271426306

**1.2. BKNet**

Tương tự như LeNet, nhưng sử dụng SGD optimizer với learning rate=0.1

**Kết quả:**

- Accuracy on public test set: 0.5371969908384534

- Accuracy on private test set: 0.5360824742600193

**2. Thử lập trình mô hình đề xuất trong bài báo dưới đây cho bài toán nhận diện cảm xúc:** <https://www.researchgate.net/profile/Dinh_Sang/publication/321257241_Facial_expression_recognition_using_deep_convolutional_neural_networks/links/5b12a7824585150a0a619d6c/Facial-expression-recognition-using-deep-convolutional-neural-networks.pdf>

Sử dụng framwork keras lập trình mô hình BKVGG12 (mô hình cho kết quả tốt nhất trích dẫn trong paper), thử nghiệm đánh giá kết quả bằng phương thức EVAL2. Quá trình training thử nghiệm sử dụng 2 hàm loss khác nhau : multi SVM loss và cross entropy loss.

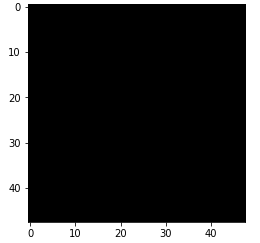
Quá trình triển khai thực hiện giống với đề xuất trong bài báo, với một số nhận xét:

**2.1. Load data**

Load data từ csv file, sử dung pandas



- Data set chứa mốt số ảnh nhiễu: toàn bộ pixels có giá trị 0. Ta loại bỏ những ảnh này khỏi training set. Các ảnh nhiễu trên traing set: 6458, 7629, 10423, 11286, 13148, 13402, 13988, 15894, 22198, 22927, 28601



**2.2. Quá trình preprocessing:**

+ Áp dụng normalize per image.

+ Tính toán mean và standard deviation per pixel trên toàn bộ training set. Lưu lại 2 ma trận (48,48) này.

+ Với mỗi ảnh bất kì trên training set hay test set, trước khi đưa vào mô hình như một đầu vào, cần được preprocess per image và preprocess per pixel sử dụng 2 gía trị mean và stddev trên.

Ảnh sau khi được preprocess:



**2.3. Data augmentation**

Sử dụng keras.preprocessing.image.ImageDataGenerator để tạo một generator với batch\_size=256. Mỗi batch\_size=256 ảnh được đưa vào, các phép biến đổi (augmentation) được áp dụng, và kết quả là 256 ảnh được đưa vào mô hình cho việc training.

- Ảnh trước khi augmentation:



- Ảnh sau khi augmentation:

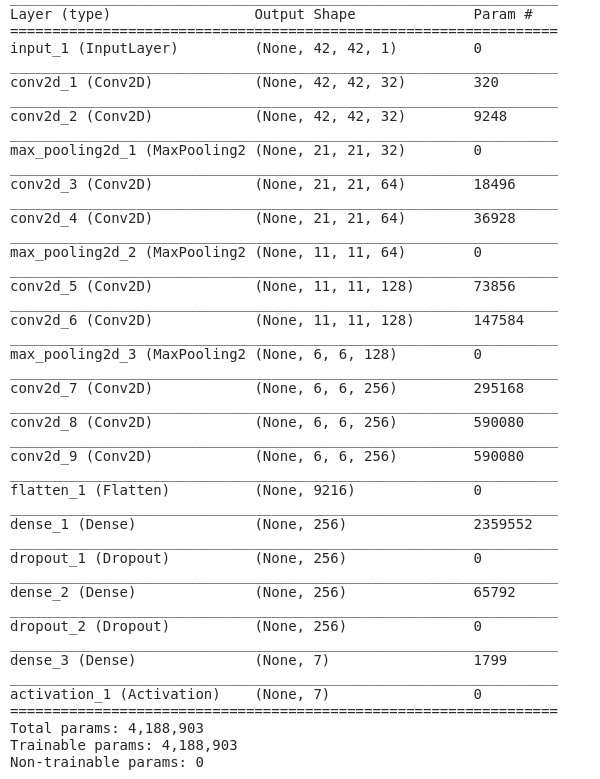


**2.4. Training mô hình**

Thử training mô hình với cross entropy loss và multiclass SVM loss.

- Tập validation sử dụng là public test set, mỗi ảnh trong đó được random augmentation tạo thành 4 ảnh. Kích thước validation set là 4\*3589.

- Minh họa: mô hình với softmax layer + cross entropy loss. Với mô hình sử dụng multiclass SVM loss, không có lớp softmax ở cuối.



- Áp dụng các callback functions:

+ ModelCheckpoint sử dung để lưu mô hình có độ chính xác cao nhất trên validation set.

+ ReduceLROnPlateau để giảm learning rate đi 10 lần khi loss không cải thiện sau 20 epochs.

+ EarlyStopping để dừng quá trình training khi loss không cải thiện sau 50 epochs.

- Tối ưu mô hình sử dụng stochastic gradient descent với learning rate=0.01 cho cross entropy loss, 0.001 cho multiclass SVM loss, momentum=0.9.

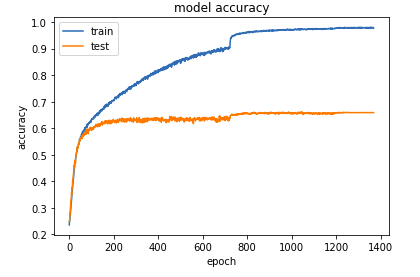
- Với multiclass SVM loss, keras chưa hỗ trợ hàm loss như trong paper. Tự lập trình hàm loss như trong paper.

- Training với tối đa 1500 epochs, batch\_size=256.

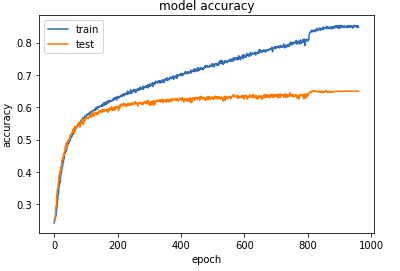
**- Kết quả:**

Biểu đồ độ chính xác trên tập train và tập validation:

+ Training history with Softmax + cross entropy loss:



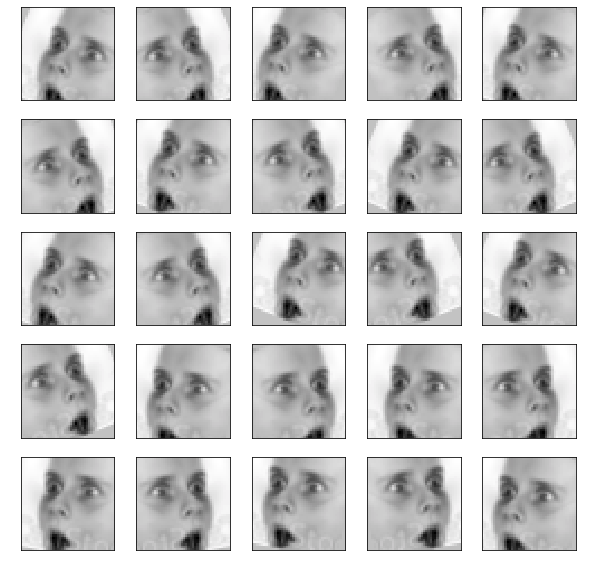
+ Training hístory with multiclass SVM loss:



**2.5. Testing**

- Một ảnh với kích thước 48\*48, trước tiên được preprocess sử dụng 2 ma trận mean và standard deviation per pixel trên training set, tính toán ở trên.

- Lập trình phương thức EVAL2 trích dẫn trong paper. Với mỗi ảnh, output ra 648 ảnh được augment theo các tham số khác nhau.



**Đánh giá mô hình theo độ chính xác:**

**- Cross entropy loss:**

+ Accuracy on public test set: **0.7010309278350515 (~70.1%)**

+ Accuracy on private test set: **0.706046252438005 (~70.6%)**

**- Multiclass SVM loss:**

+ Accuracy on public test set: **0.6990805238227918 (~69.9%)**

+ Accuracy on private test set: **0.7015881861242687 (~70.2%)**

Kết quả thu được còn cách xa kết quả trong paper. Đặc biệt, khác với trích dẫn trong paper, phương pháp L2 multiclass SVM loss cho kết quả không tốt bằng cross entropy loss với sofmax layer ở cuối.

**3. Tìm hiểu kỹ thuật sử dụng các mô hình mạng CNN kinh điển như VGG, ResNet, DenseNet với framework Keras, thử áp dụng cho các bài toán nhận diện cảm xúc, phân loại phương tiện giao thông:** [**https://keras.io/applications/**](https://keras.io/applications/)

- Keras cung cấp các mô hình kinh điển áp dụng với tập dữ liệu ImageNet với 1000 class. Các mô hình này có thể được sử dụng như là một base model cho các bài toán object classification.

- Các lớp convolution, pooling,.. trong các mô hình kinh điển này có khả năng trích xuất đặc trưng rất mạnh. Vì vậy, có thể linh hoạt tùy chỉnh, tái sử dụng cho việc trích xuất đặc trưng, khởi tạo weights, hay đánh giá các base model.

**- Kĩ thuật fine-tune:**

+ B1: Lựa chọn một base model là một mô hình kinh điển, phù hợp và gần với bài toán đang xét.

+ B2: Xây dựng mô hình trong đó sử dụng các lớp convolution, pooling của các base model cho việc trích xuất đặc trưng. Cụ thể, loại bỏ các fully-connected layer gần output và thay thể bởi các lớp phù hợp cho bài toán đang xét.

+ B3: Đóng băng tham số (weights) của các convolution layers. Nói cách khác, ta coi như base model là một bộ trích chọn đặc trưng tốt, giữ lại các weight ở các layers này, chỉ huấn luyện tham số cho các lớp fully-connected.

+ B4: Huấn luyện tham số cho bộ phân loại (các lớp fully-connected)

+ B5: Lựa chọn các lớp cần điều chỉnh tham số, phá băng cho các lớp này, huấn luyện mô hình với learning rate nhỏ, với mong muốn giữ lại khả năng trích xuất đặc trưng mạnh, tham số cho các lớp thuộc base model.

**3.1. Thử nghiệm với bài toán phân loại phương tiện giao thông**

Thử nghiệm với base model là ResNet50 và VGG16. Trong cả 2 trường hợp, thêm vào sau 3 fully-connected layer với số units là 512,64 và 5. Layer cuối kết hợp softmax.

**3.2.1. ResNet50**

- Sau khi đóng băng các lớp base model, huấn luyện với optimizer là adam, 20 epochs.

- Train tham số của toàn bộ mô hình với optimizer là Stochastic gradient descent, learning rate là 0.001, epochs=100, batch\_size=64.

Kết quả, độ chính xác trên tập test là 0.9016393432851697

**3.2.2. VGG16 kết hợp data augmentation**

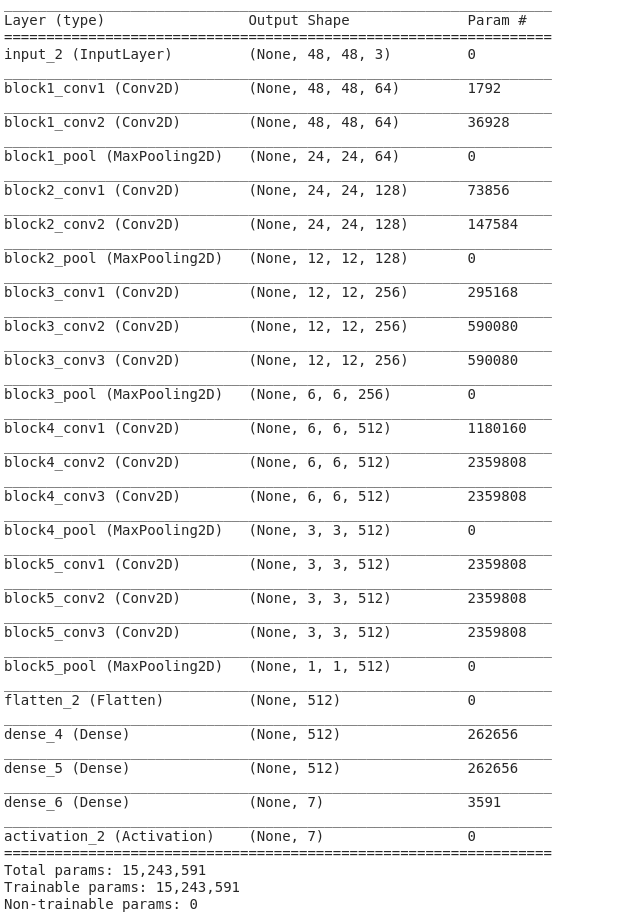
Tương tự như trên, chỉ khác là sử dụng thêm data augmentation để tăng lượng training set, tránh overfitting.

Cụ thể, dữ liệu được random rotate các góc (-30,30) độ, mirror, điều chỉnh sáng, nomalize về khoảng [0,1].

Kết quả, độ chính xác trên tập test là 0.9016393406795022

**3.2. Thử nghiệm với bài toán phân loại cảm xúc FERC-2013**

Thử nghiệm với base model là VGG16. Thêm 3 lớp fully-connected layers với số units là 512,512,7 kết hợp softmax ở layer cuối cùng.



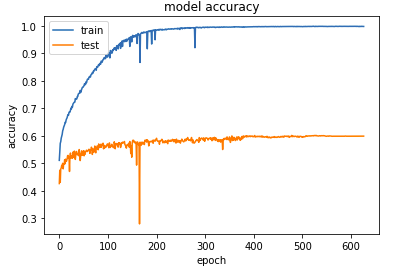
- FERC-2013 là tập dữ liệu trong đó ảnh có shape (48,48,1), tức là số channel chỉ là 1. Trong khi đó, các pretrain model chỉ chấp nhận input là ảnh 3 channels. Giải quyết bằng cách mở rộng ảnh gốc 1 channel thành 3 channels, với giá trị mỗi channel tại từng điểm ảnh là giống nhau.

- Áp dụng augmentation trên training set, đưa vào mô hình để huấn luyện.

- Tập validation sử dụng chính là tập public test set, sử dụng để đánh giá mô hình.

Kết qủa:

Biểu đồ accuracy trong quá trình train (toàn bộ layers):



**Độ chính xác:**

- Trên tập public test: 0.5987740317969377

- Trên tập private test: 0.6068542769905849

Độ chính xác được cải thiện đáng kể so với mô hình ở bài tập 1.