IDL 머신러닝에 의한 필기숫자 분류

ENVI and IDL are trademarks of L3HARRIS, Inc.
http://www.harrisgeospatial.com

Intro

머신러닝(Machine Learning)은 IT 분야에서 이미 대세가 되었습니다, IDL에서는 8.7.1 버전에서부터 이 기능이 탑재되기 시작했습니다. 이번 한장강의에서는 머신러닝 분야에서 널리 알려진 MNIST 데이터베이스를 활용하여 필기 숫자를 분류하는 모델 소개해 보겠습니다. IDL의 머신러닝 기능을 활용하여 구현한 예제를 관련 프로그램과 함께 소개해보고자 합니다.

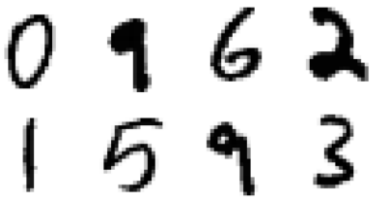
예제 프로그램 다운로드

소개를 위하여 사용할 예제 프로그램은 classify_digits_lsw입니다. 원래는 classify_digits라는 이름으로 IDL 8.7 및 8.8에서 기본 제공되는 프로그램인데, 몇가지 개선을 거쳐서 수정한 버전입니다. classify_digits_lsw.pro 파일은 아래 링크를 통하여 다운로드받을 수 있습니다.

다운로드 링크 누르기

MNIST 데이터 베이스란?

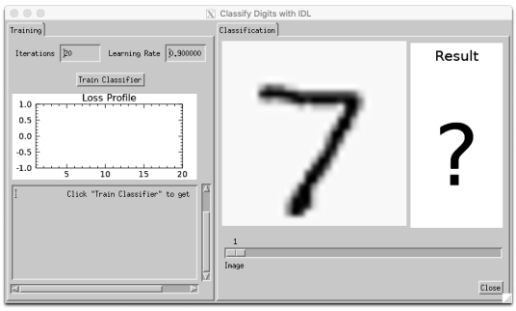
MNIST(Modified National Institute of Standards and Technology) 데이터베이스는 머신러닝 분야에서 훈련 및 테스트용으로 널리 사용되는 데이터로서, 여러 종류가 있지만 그 중 사람이 손으로 쓴 0~9의 숫자들에 대한 이미지들로 구성된 데이터가 있습니다. 이 데이터는 실제로는 6만개의 훈련용(Training) 이미지들과 1만개의 평가용(Validation) 이미지들로 구성되어 있는데, 각 이미지는 28x28의 구조를 갖는 바이트스케일 이미지이며, 6만개의 훈련용 이미지들 중 몇가지만 보면 다음 그림과 같습니다.



그래서 머신러닝 분야에서는 사람이 손으로 쓴 0~9의 숫자를 인식하여 판별하는 모델을 제작하는데 있어서 이 MNIST 데이터를 사용하는 경우가 많으며, 특히 머신러닝에 대한 교육 등에서 이러한 예제가 많이 활용됩니다.

예제 프로그램의 사용법

예제 프로그램 파일인 classify_digits_lsw.pro을 받아서 컴파일 및 실행을 해보면, 다음 그림과 같이 독립적인 GUI를 갖는 어플리케이션의 형태로 실행됩니다.



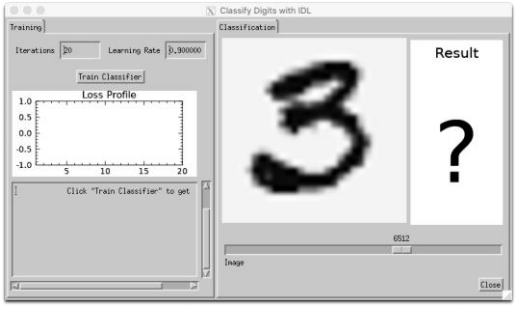
이 프로그램은 IDL 8.7.1 또는 그 이후 버전의 IDL에서만 정상적으로 작동하며, 위 그림은 Mac OS용 IDL 8.8에서 실행했을 때의 모습입니다. Windows용 IDL에서 실행한 모습도 내부 그림상의 문자 폰트를 제외하면 같습니다.

그리고 이 프로그램은 시작과 동시에 MNIST 데이터베이스 파일들을 자동으로 다운로드합니다. 따라서 정상적인 실행을 위해서는 네트워크 연결이 반드시 필요합니다. 참고로 이 프로그램이 데이터 다운로드를 위하여 접속하는 곳은 IDL Git 허브에 있는 mnist-data 디렉토리이며 링크는 다음과 같습니다. 물론 여러분들이 직접 이 링크로 굳이 가실 필요는 없습니다.

https://github.com/interactive-data-language/mnist-data

Step 1. 평가용 이미지 조회

먼저 UI상에서 우측 하단에 있는 슬라이드바를 조정해봅시다. 이 슬라이드바는 1~10000의 범위를 갖습니다. 즉 MNIST 데이터에 수록된 1만개의 평가용 이미지들을 하나씩 화면상에서 볼 수 있도록 해주는 역할을 합니다. 예를 들어 번호가 6512가 되도록 슬라이드바를 위치시킨 모습은 다음과 같습니다.

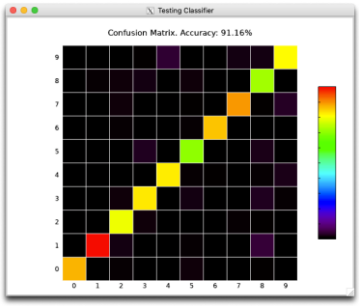


아마도 숫자 3을 필기로 쓴 모습인 것 같습니다. 물론 아직 모델을 훈련시키기 전이기 때문에 결과는 나오지 않은 상태입니다. 그래서 바로 우측에 있는 결과창에서는 ?로만 표시되어 있습니다. 잠시 후 훈련이 끝난 후에는 이 ?가 모델의 판독 결과로 바뀌게 될 것입니다.

Step 2. 모델의 훈련

이제 모델 훈련을 시작하기 위해서는 UI상의 좌측상단에 있는 'Train Classifier'라는 버튼을 누르면 됩니다. 훈련을 통하여 얻게 되는 모델은 결국 임의의 필기 숫자에 대하여 0~9 범위의 총 10가지 숫자들 중 하나로 판별하게 됩니다. 즉 분류(Classification) 목적의 모델을 구축하는 셈입니다. 그리고 이러한 분류형 모델을 구축하는데 있어서는 여러가지 세부 기법들이 존재하는데 IDL의 머신러닝 기능에서는 Support Vector Machine, SoftMax, Feed Forward Neural Network 등 세가지가 지원됩니다. 지금의 예제 프로그램에서는 SoftMax 기법이 적용됩니다. 그리고 한번의 실행에 의하여 훈련이 종료되는 SVM과는 달리 SoftMax의 경우는 일정 횟수의 반복(Iteration) 과정을 거치게 됩니다.

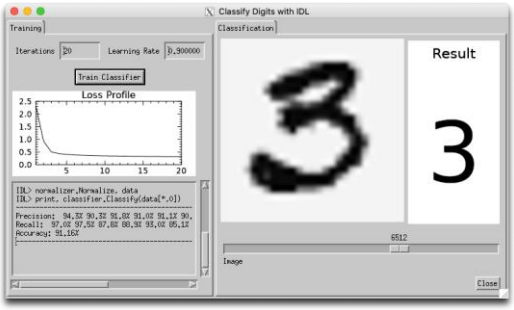
'Train Classifier' 버튼의 위쪽을 보면, 훈련과정의 반복 횟수를 결정하는 Iterations, 그리고 학습률을 결정하는 Learning Rate라는 항목들이 보입니다. 물론 이 항목들의 값은 수정이 가능하지만, 일단은 미리 수록되어 있는 20과 0.9라는 값을 그대로 사용합시다. 버튼을 눌러서 훈련과정이 시작되면 약간의 시간이 소요됩니다. 그리고 그 과정에서 별도의 팝업 그래픽창이 뜨면서 다음과 같은 그림이 보일 것입니다.



이 그림은 Confusion Matrix를 나타냅니다. 이 예제 프로그램 내의 훈련과정을 보면, 6만개의 훈련용 데이터 중에서 90%인 5만4천개가 훈련용으로 사용되고 10%인 6천개는 검증용(Test)으로 사용됩니다. 즉 90%의 데이터를 SoftMax 방법으로 훈련시켜서 모델을 구축한 다음, 이 모델을 10%의 검증용 데이터에 적용해서 0~9 각 숫자별 적중률을 나타낸 것이 Confusion Matrix라고 보시면 됩니다. 그리고 훈련 과정에서 반복 횟수가 늘어날수록 정확도가 증가하며, 이 그림의 상단에서 Accuracy라고 표시됩니다. 이 그림에서는 정확도가 91.16%라고 표시되었지만, 이 값은 훈련을 할 때마다 달라질 수 있습니다.

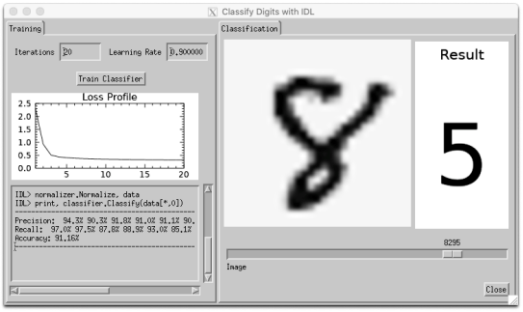
Step 3. 모델의 성능 확인

훈련이 종료된 후 메인 UI를 보면 그 모습은 다음과 같습니다.



여기서 좌측 중간쯤에 있는 Loss Profile이라는 제목의 그래프가 보이는데, 훈련된 모델이 데이터와 얼마나 잘 맞느냐를 나타내는 Loss라는 양이 반복 과정에서 어떻게 변하는가를 나타냅니다. 이 값이 작을수록 모델이 데이터를 잘 기술한다는 의미이며, 반복을 거쳐 그 값이 계속 감소하였음을 나타냅니다. 즉 훈련이 잘 진행되었다는 의미로 해석하면 됩니다.

그리고 위의 그림을 보면 우측에 있는 Result 창에 이제는 ? 대신 3이라는 숫자가 적혀있는 것이 보입니다. 즉 바로 왼쪽에 있는 평가용 이미지에 대하여 모델이 판단한 숫자값이 3이란 의미입니다. 사람이 보는 시각과 일치하는 결과라고 볼 수 있습니다. 물론 이 모델의 정확도가 100%는 아니기 때문에 다음과 같이 우리의 생각과 일치하지 않는 경우들도 일부 존재합니다.



물론 모델의 성능 및 정확도는 훈련의 과정에서 어떤 기법을 사용하느냐 그리고 훈련용 데이터를 어떤 식으로 활용하느냐 등에 따라 얼마든지 달라질 수 있습니다. 여기서는 예제 프로그램을 통하여 그러한 예들 중 하나를 제시한 것 뿐이라고 봐야 합니다.

IDL의 머신러닝에 관한 참고자료

IDL의 머신러닝 기능에 대한 내용은 IDL 도움말의 검색창에서 machine learning으로 검색하여 표시되는 항목들 중 The IDL Machine Learning Framework라는 제목의 페이지를 보면 예제들과 함께 구현 과정이 잘 설명되어 있습니다. 또한 이 내용을 바탕으로 작성된 블로그 게시물들도 있으므로 함께 참조하시면 좋을 것 같습니다.

http://blog.daum.net/swrush/558