예측애널리틱스 Homework #7

고려대학교
KOREA UNIVERSITY

대학	고려대학교 공과대학
학과	산업경영공학부
학번	2017170819
이름	박상민

1. Convolutional neural network 를 구축하고 분류 성능을 일반 deep neural network 와 비교하세요.
   (CNN 에서 convolution layer, pooling layer, fiter 사이즈, filter 수 등은 여러분들이 변경해 가면서 해 보세요. 기존 구조 (Alexnet, VGG, ResNet 등)을 이용해도 좋습니다)

데이터의 크기가 매우 크기 때문에 1/10 만 샘플링해서 모델링을 진행했다. Training data 는 모델은 구축하는 데에만 사용되고, 파라미터 튜닝에 사용되는 데이터는 validation data 이다.

먼저 CNN 과 DNN 을 구축했다. 입력 노드=500 개 , 은닉층=2, 은닉노드=300, 100 활성화함수=Relu(1), Leaky Relu(2)

1) Convolution1: 3 by 3 의 filter 10 개

   Convolution2: 3 by 3 의 filter 20 개

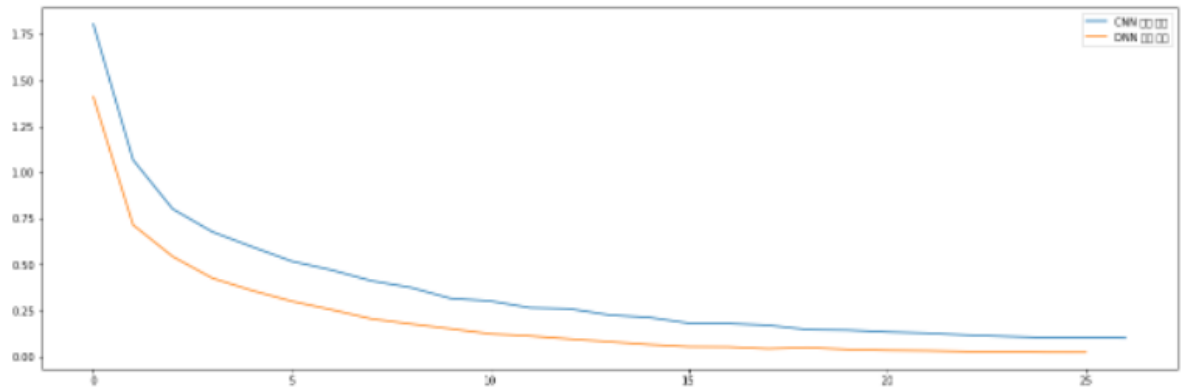
   Convolution3: 3 by 3 의 filter 40 개

   Pooling layer: 2 by 2, stride=2

CNN 을 구현한 결과, 총 138 초가 소요됐고, DNN 은 87 초가 소요됐다.

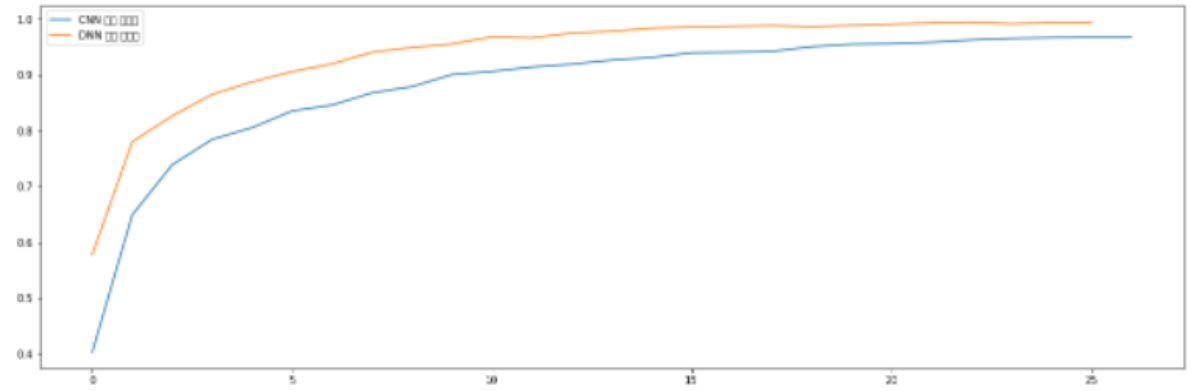
CNN 학습 로스

DNN 학습 로스



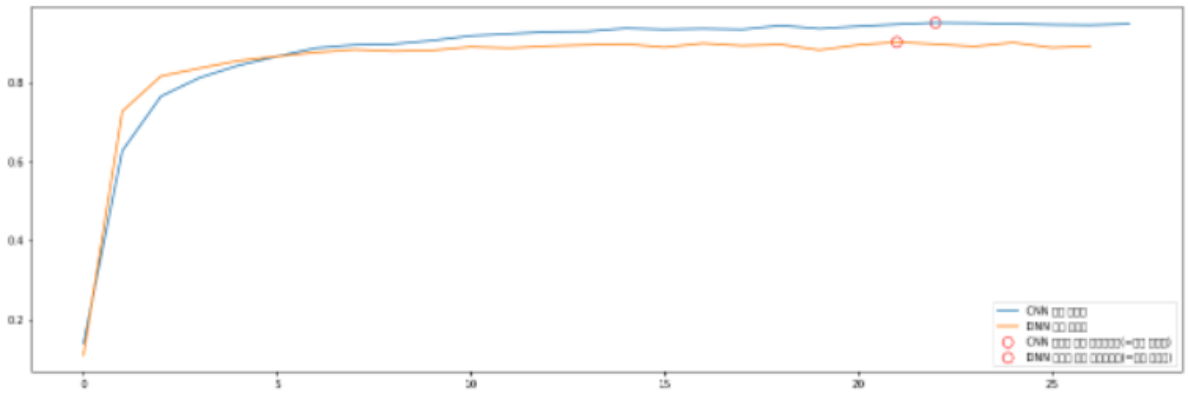
CNN 학습 정확도

DNN 학습 정확도



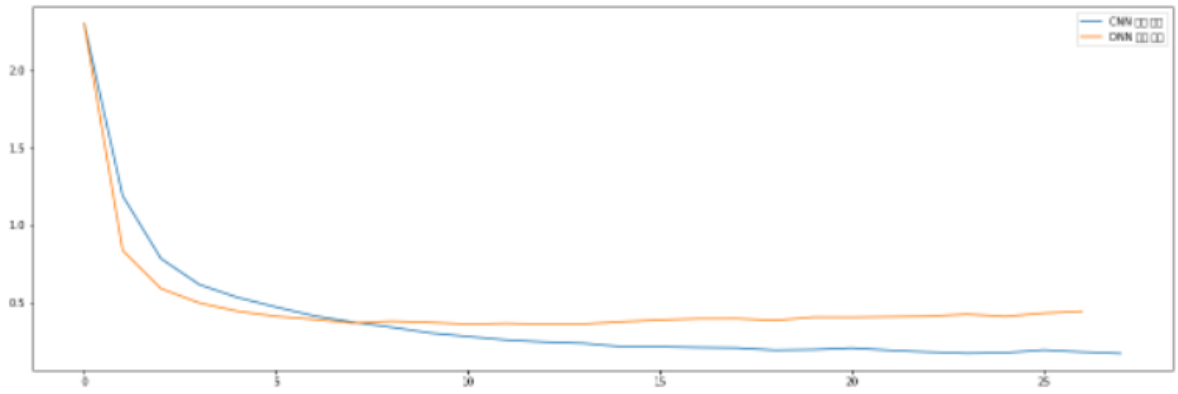
CNN 검증 정확도

DNN 검증 정확도



CNN 검증 로스

DNN 검증 로스

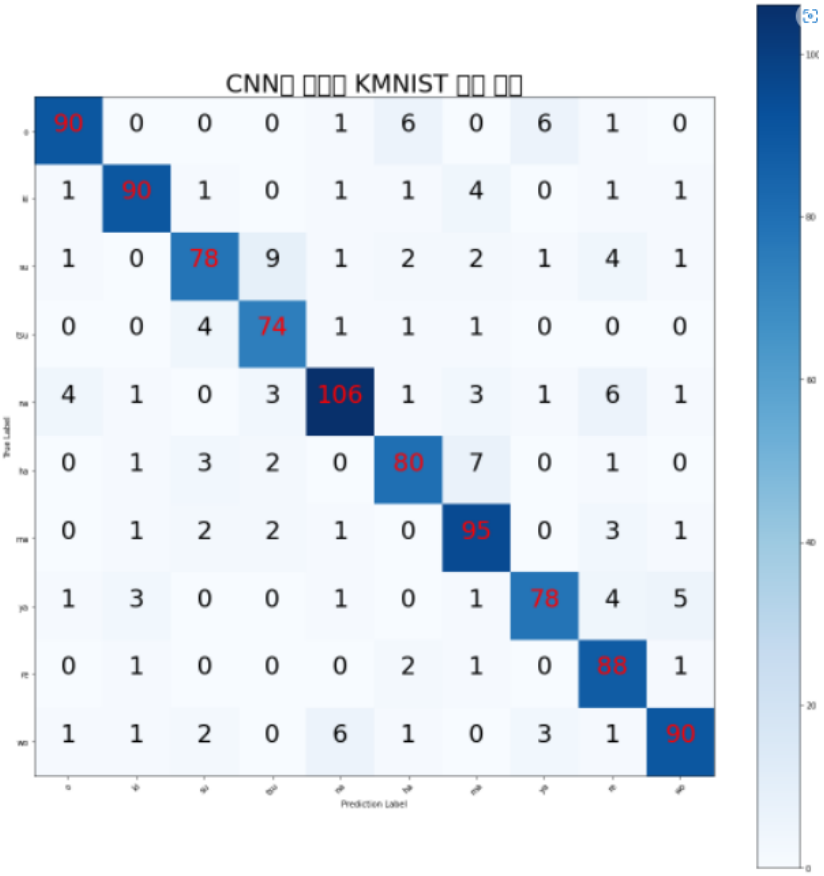


(22, 21)

CNN 은 총 26 번의 epoch 가 실행됐고, DNN 은 총 25 번의 epoch 가 실행됐다. epoch 가 반복될수록 training loss 와 validation loss 가 감소한 것을 확인할 수 있다. Training accuracy 를 비교해보면 DNN 의 학습 정확도가 CNN 의 학습 정확도보다 더 높은 것을 확인할 수 있다. 하지만 validation accuracy 를 비교해보면, CNN 이 DNN 보다 더 높은 것을 확인할 수 있다. 따라서 DNN 은 CNN 보다 overfitting 되었다고 할 수 있다. 손실함수도 CNN 이 DNN 보다 더 작기 때문에 더 좋은 예측력을 보였다고 할 수 있다. 이때 검증 정확도가 가장 높았던 지점에서 최적의 학습 체크포인트(학습중단점)를 나타냈으며, 이때가 가장 효과적인 예측을 한 지점이다. CNN 은 22 번째 epoch, DNN 은 21 번째 epoch 가 최적의 학습 중단점이었다.

```
CNN 분류 정확도: 0.869 | DNN 분류 정확도: 0.792
```

이제 파라미터 튜닝을 완료했기 때문에 test accuracy 를 확인해보았다. 정확도를 도출한 결과, CNN 의 분류 정확도는 0.869, DNN 의 분류 정확도는 0.792 를 기록했다. 따라서 CNN 의 분류 성능이 DNN 보다 월등히 좋음을 확인할 수 있었다.



실제 어떻게 데이터 관측치들이 classification 되었는지를 보여주는 confusion matrix 를 그려보았다. 실제 label 에 대해 10 class 모두 골고루 잘 예측되었음을 확인할 수 있다.

2) Convolution1: 3 by 3 의 filter 10 개

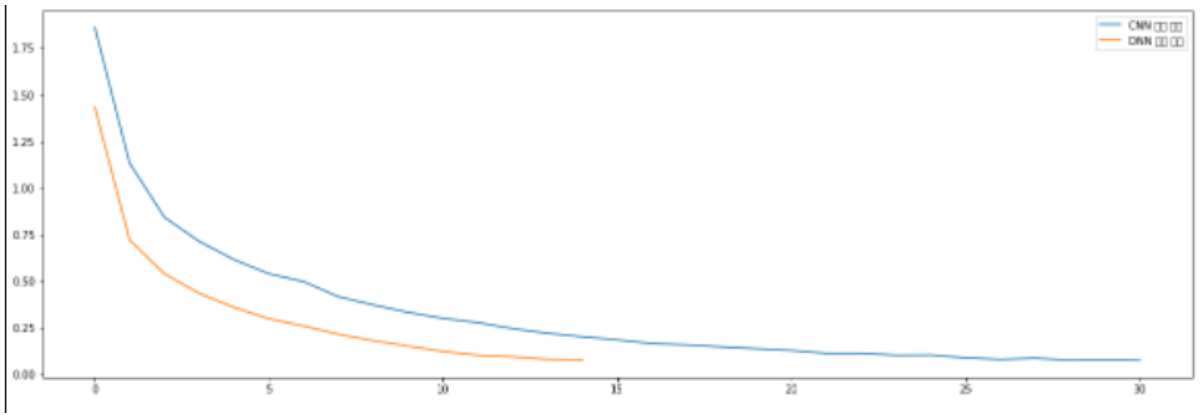
Convolution2: 3 by 3 의 filter 20 개

Pooling layer: 2 by 2, stride=2

Convolution3 과정을 생략하는 방향으로 유저 파라미터를 바꿔보았다. CNN 은 184 초, DNN 은 56 초가 소요됐다.

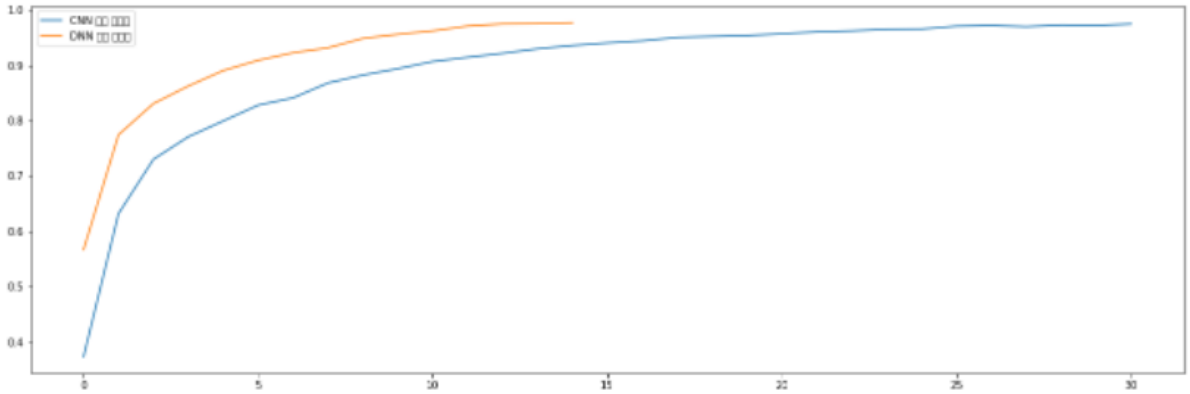
CNN 학습 로스

DNN 학습 로스



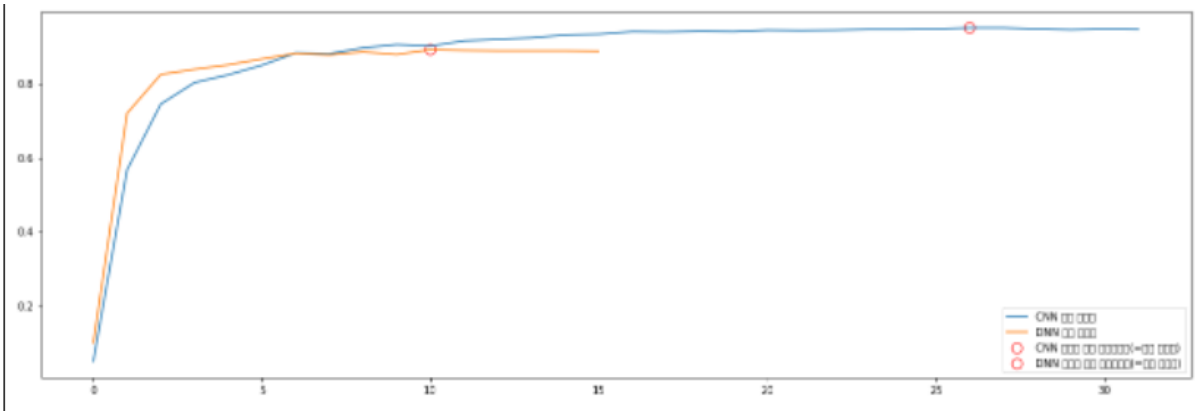
CNN 학습 정확도

DNN 학습 정확도



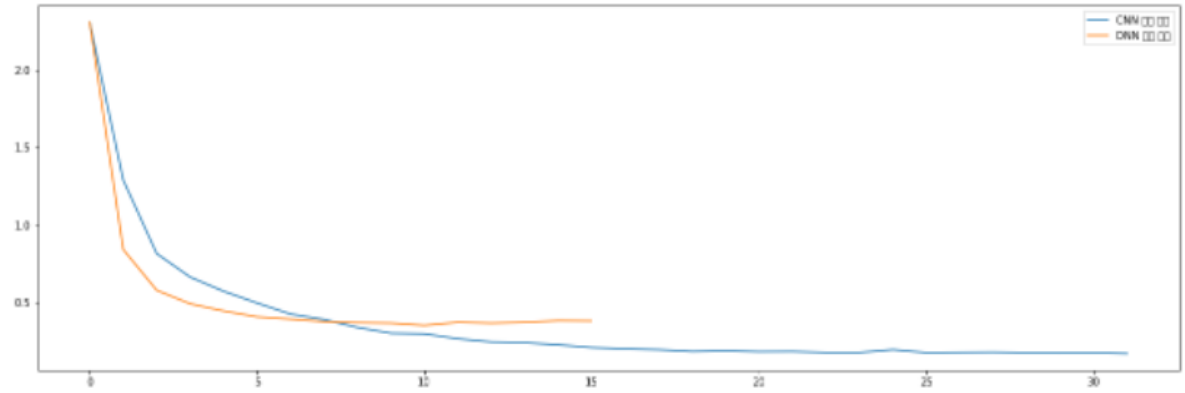
CNN 검증 정확도

DNN 검증 정확도



CNN 검증 로스

DNN 검증 로스

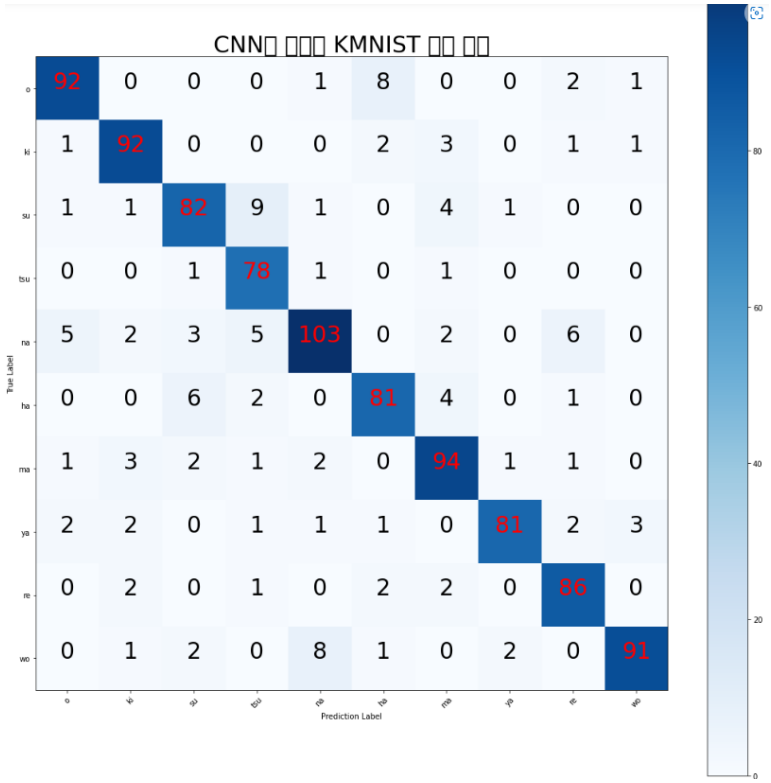


(26, 10)

CNN 은 총 30 번의 epoch 가 실행됐고, DNN 은 총 14 번의 epoch 가 실행됐다. epoch 가 반복될수록 training loss 와 validation loss 가 감소한 것을 확인할 수 있다. Training accuracy 를 비교해보면 DNN 의 학습 정확도가 CNN 의 학습 정확도보다 더 높은 것을 확인할 수 있다. 하지만 validation accuracy 를 비교해보면, CNN 이 DNN 보다 더 높은 것을 확인할 수 있다. 따라서 DNN 은 CNN 보다 overfitting 되었다고 할 수 있다. 손실함수도 CNN 이 DNN 보다 더 작기 때문에 더 좋은 예측력을 보였다고 할 수 있다. 이때 검증 정확도가 가장 높았던 지점에서 최적의 학습 체크포인트(학습중단점)를 나타냈으며, 이때가 가장 효과적인 예측을 한 지점이다. CNN 은 26 번째 epoch, DNN 은 10 번째 epoch 가 최적의 학습 중단점이었다.

```
CNN 분류 정확도: 0.880 | DNN 분류 정확도: 0.809
```

이제 파라미터 튜닝을 완료했기 때문에 test accuracy 를 확인해보았다. 정확도를 도출한 결과, CNN 의 분류 정확도는 0.884, DNN 의 분류 정확도는 0.782 를 기록했다. 따라서 CNN 의 분류 성능이 DNN 보다 월등히 좋음을 확인할 수 있었다.



실제 어떻게 데이터 관측치들이 classification 되었는지를 보여주는 confusion matrix 를 그려보았다. 실제 label 에 대해 10 class 모두 골고루 잘 예측되었음을 확인할 수 있다

3) Convolution1: 3 by 3 의 filter 10 개

Convolution2: 3 by 3 의 filter 20 개

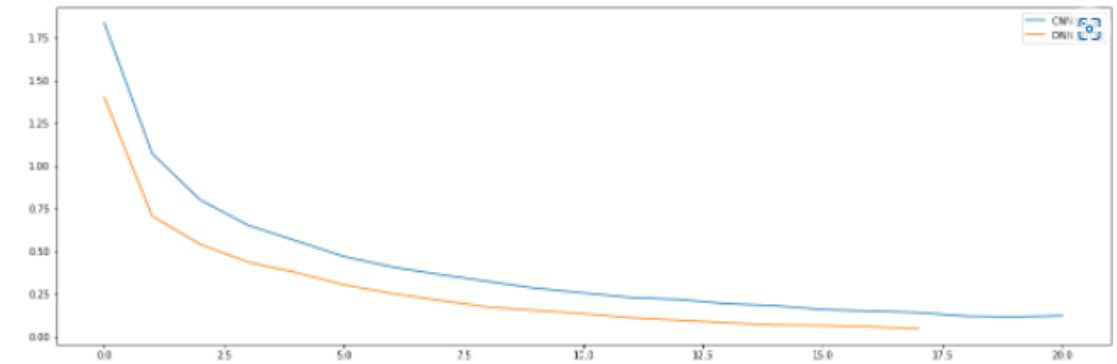
Convolution3: 3 by 3 의 filter 20 개

Pooling layer: 2 by 2, stride=2

Convolution3 과정의 필터 개수를 증가시키는 방향으로 유저 파라미터를 바꿔보았다. CNN 은 118 초, DNN 은 63 초가 소요됐다.

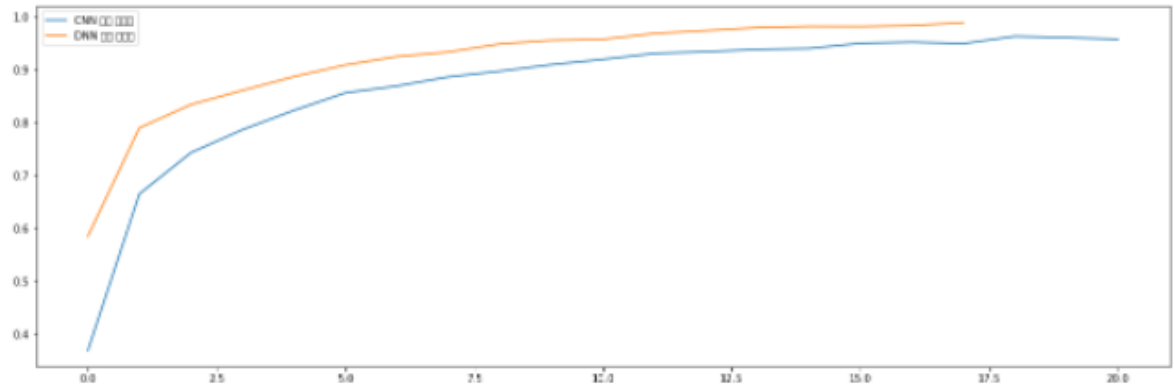
CNN 학습 로스

DNN 학습 로스



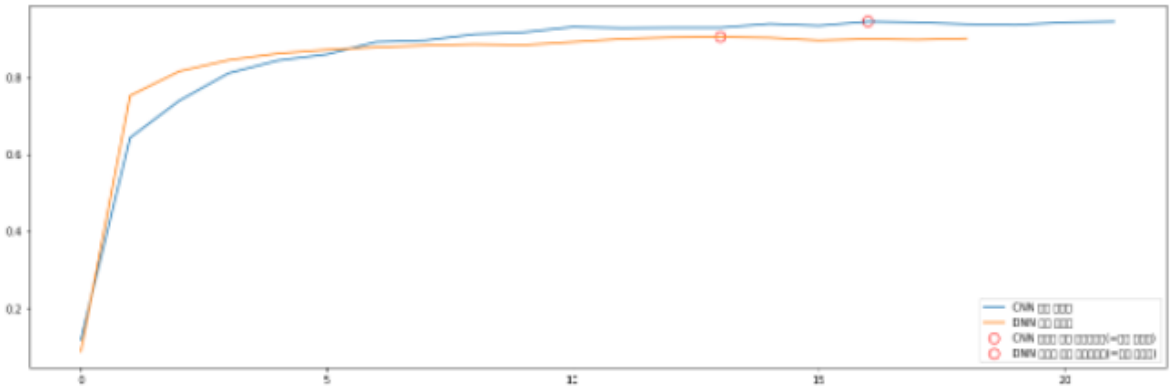
CNN 학습 정확도

DNN 학습 정확도



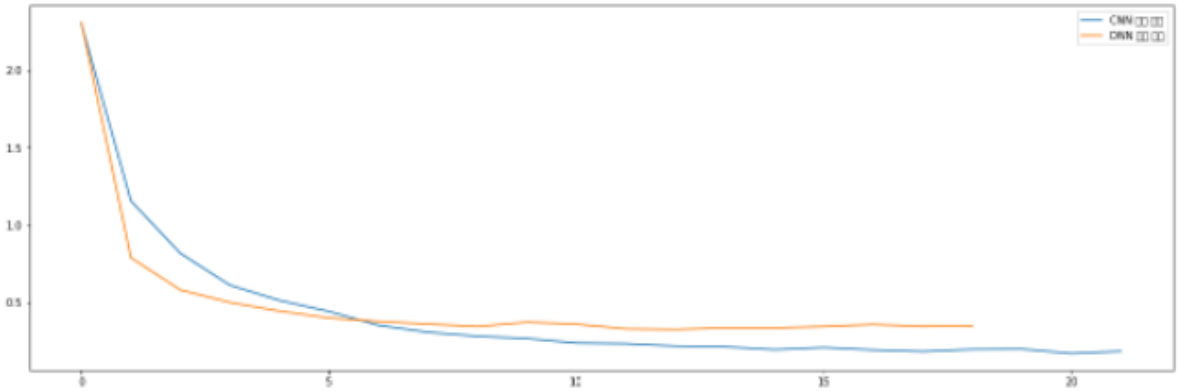
CNN 검증 정확도

DNN 검증 정확도



CNN 검증 로스

DNN 검증 로스



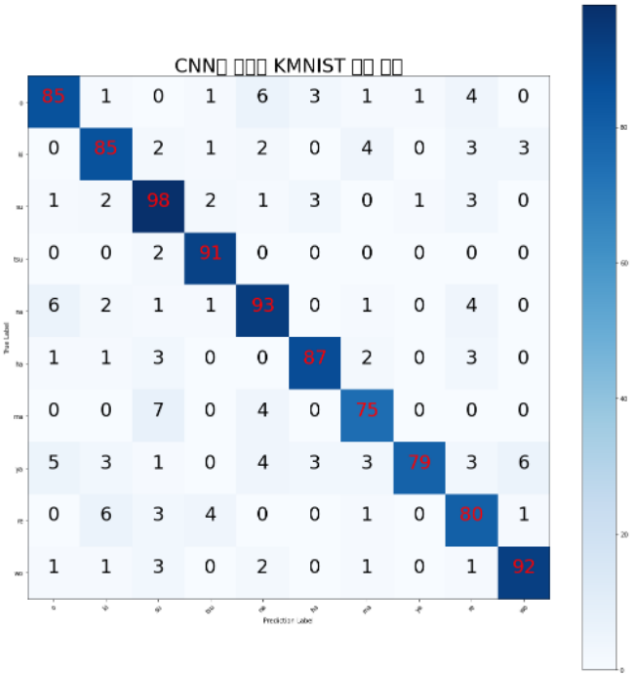
(16, 13)

CNN 은 총 20 번의 epoch 가 실행됐고, DNN 은 총 17 번의 epoch 가 실행됐다. epoch 가 반복될수록 training loss 와 validation loss 가 감소한 것을 확인할 수 있다. Training accuracy 를 비교해보면 DNN 의 학습 정확도가 CNN 의 학습 정확도보다 더 높은 것을 확인할 수 있다. 하지만 validation accuracy 를 비교해보면, CNN 이 DNN 보다 더 높은 것을 확인할 수 있다. 따라서 DNN 은 CNN 보다 overfitting 되었다고 할 수 있다. 손실함수도 CNN 이 DNN 보다 더 작기 때문에 더 좋은 예측력을 보였다고 할 수 있다. 이때 검증 정확도가 가장 높았던 지점에서 최적의 학습 체크포인트(학습중단점)를 나타냈으며, 이때가 가장 효과적인 예측을 한 지점이다. CNN 은 16 번째 epoch, DNN 은 13 번째 epoch 가 최적의 학습 중단점이었다.

필터의 종류 개수가 감소했기 때문에 그만큼 feature map 의 수도 작아진다. 따라서 그만큼 파라미터 튜닝도 덜 진행된다고 할 수 있다. 튜닝된 파라미터가 최적의 파라미터가 아닐 수 있고, 예측 성능도 case 1 보다 떨어질 가능성이 있다.

```
CNN 분류 정확도: 0.865 | DNN 분류 정확도: 0.790
```

분류정확도를 계산해본 결과, case1 과 정확도가 거의 비슷했지만, 약간 낮은 것을 볼 수 있다. 따라서 생성된 feature map 의 개수가 살짝 부족해서 파라미터 튜닝이 완벽하게 되지 못했다고 결론지을 수 있다. 역시 마찬가지로 DNN 이 CNN 보다 overfitting 되어있기 때문에 CNN 의 분류 정확도가 월등히 높았다.



실제 어떻게 데이터 관측치들이 classification 되었는지를 보여주는 confusion matrix 를 그려보았다. 실제 label 에 대해 10 class 모두 골고루 잘 예측되었음을 확인할 수 있다

4) Convolution1: 3 by 3 의 filter 10 개

Convolution2: 3 by 3 의 filter 20 개

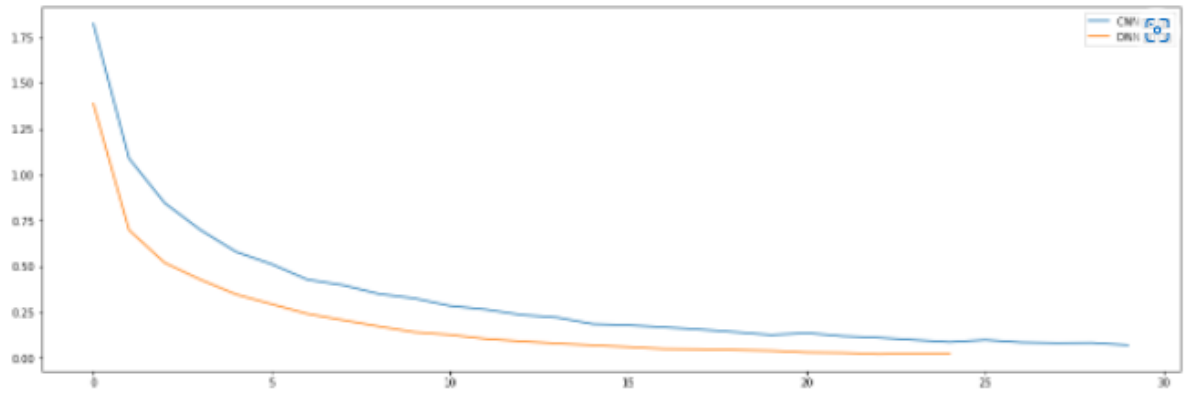
Convolution3: 5 by 5 의 filter 40 개

Pooling layer: 2 by 2, stride=2

Convolution3 의 필터 사이즈를 5 by 5 로 크게 키워보았다.

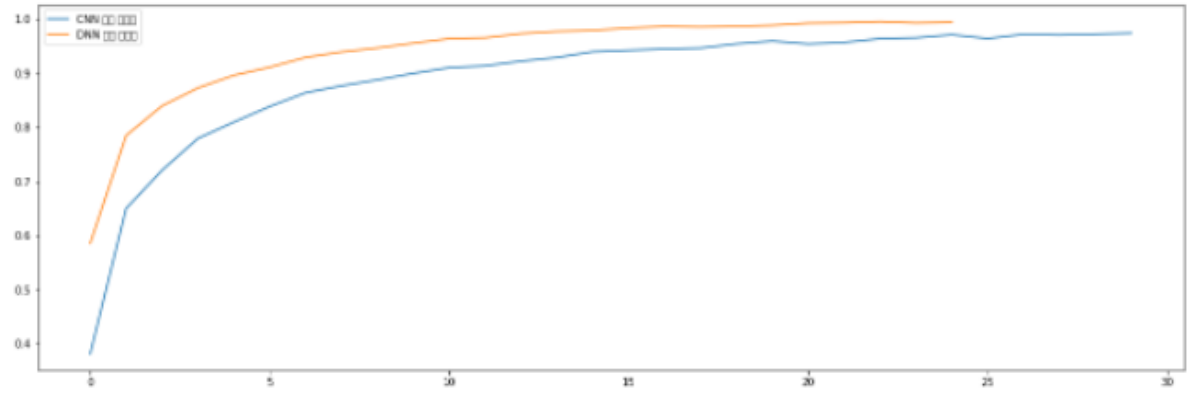
CNN 학습 로스

DNN 학습 로스



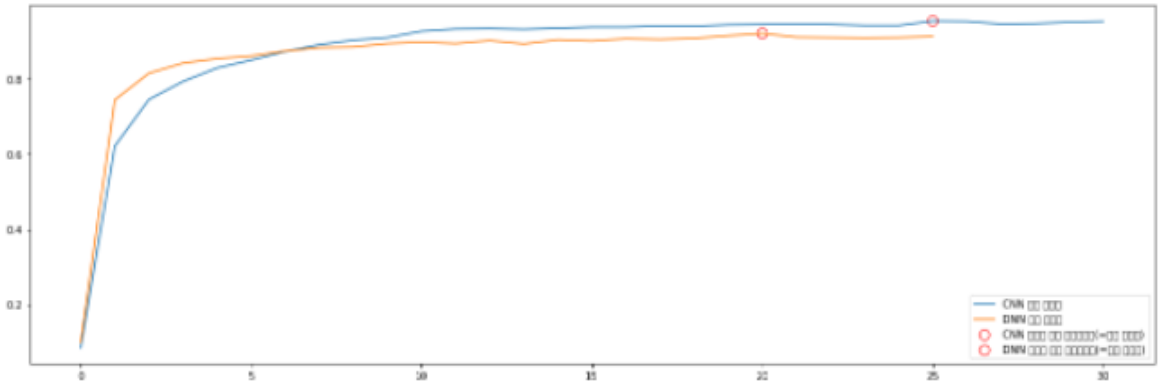
CNN 학습 정확도

DNN 학습 정확도



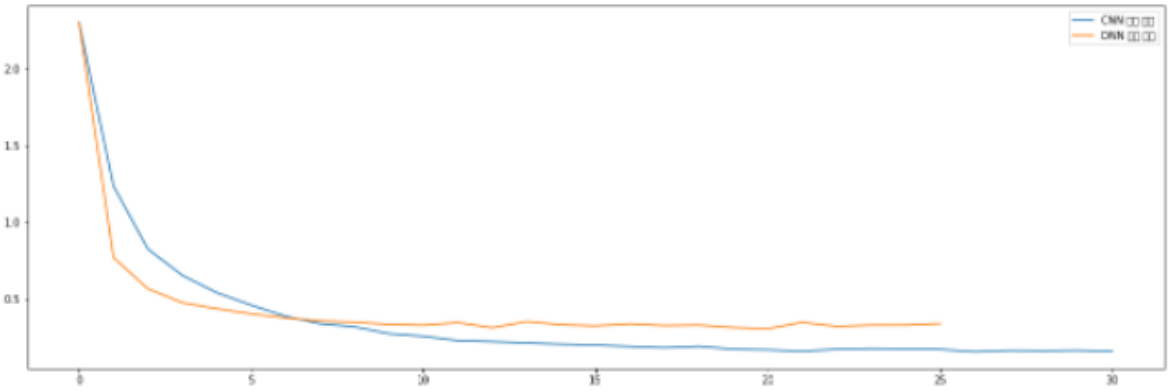
CNN 검증 정확도

DNN 검증 정확도



CNN 검증 로스

DNN 검증 로스



```
(25, 20)
```

CNN 은 총 29 번의 epoch 가 실행됐고, DNN 은 총 24 번의 epoch 가 실행됐다. epoch 가 반복될수록 training loss 와 validation loss 가 감소한 것을 확인할 수 있다. Training accuracy 를 비교해보면 DNN 의 학습 정확도가 CNN 의 학습 정확도보다 더 높은 것을 확인할 수 있다. 하지만 validation accuracy 를 비교해보면, CNN 이 DNN 보다 더 높은 것을 확인할 수 있다. 따라서 DNN 은 CNN 보다 overfitting 되었다고 할 수 있다. 손실함수도 CNN 이 DNN 보다 더 작기 때문에 더 좋은 예측력을 보였다고 할 수 있다. 이때 검증 정확도가 가장 높았던 지점에서 최적의 학습 체크포인트(학습중단점)를 나타냈으며, 이때가 가장 효과적인 예측을 한 지점이다. CNN 은 16 번째 epoch, DNN 은 13 번째 epoch 가 최적의 학습 중단점이었다.

필터의 사이즈가 증가했기 때문에 그만큼 뉴럴네트워크의 가중치들, 즉 파라미터가 많아진다. 튜닝할 파라미터가 많아진 만큼 파라미터 튜닝을 위해 더 많은 데이터가 필요하다. 따라서 모델의 파라미터가 최적의 파라미터가 아닐 수 있고, 예측 성능도 case 1 보다 떨어질 가능성이 있다.

```
CNN 분류 정확도: 0.861 | DNN 분류 정확도: 0.789
```

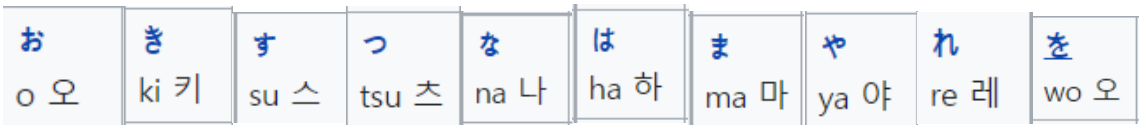
분류정확도를 계산해본 결과, case1 과 정확도가 거의 비슷했지만, 약간 낮은 것을 볼 수 있다. 따라서 파라미터의 튜닝이 case1 에 비해 정확하지 못해서 case1 보다 정확도가 낮았다. 역시 마찬가지로 DNN 이 CNN 보다 overfitting 되어있기 때문에 CNN 의 분류 정확도가 월등히 높았다.

2. 위 CNN 결과에 CAM 과 Grad-CAM 을 적용하여 각 클래스별로 주요 영역을 표시하기 바랍니다.

CAM(Class Activation Map)은 이미지의 어떤 부분이 이미지의 클래스 예측에 중요한지 히트맵으로 나타내는 방법이다. CNN 의 성능을 다른 머신러닝 기법들에 비해 매우 좋다. 하지만 blackbox model 이기 때문에 왜 이 관측치의 결과가 이렇게 나왔는지 설명하지 못한다. 따라서 이미지의 어떤 부분을 보고 클래스를 결정한 것인지 알 필요가 있다.

CAM 의 원리는 마지막 convolutional layer 뒤에 global average pooling 구조를 사용해 중요한 부분에 가중치가 곱해질 수 있도록 하는 것이다. 각 feature map 들을 각 숫자 하나로 대표할 수 있도록 각 feature map 의 요소값의 평균을 구한다. 그 다음 이 값들을 뉴럴네트워크의 input 으로 넣어 파라미터를 학습한다. 그리고 각 클래스를 예측할 때 사용되는 가중치들을 feature map 에 곱해 가중치가 적용된 feature map 을 만든다. 따라서 이 feature map 들을 하나의 이미지로 합하면, 이 클래스로 예측하게 된 요인이 된 부분이 집중적으로 밝게 나타난다.

Grad-CAM 은 CAM 의 구조적인 한계점으로 보완하기 위해 나왔다. CAM 은 global average pooling layer 를 반드시 사용해야 했기 때문에 뒷부분에 대한 또 다시 fine tuning 을 해야 했다. 따라서 마지막 convolutional layer 에 대해서만 CAM 추출이 가능했다. 이를 보완하기 위해 global average pooling 을 사용하지 않고, CNN 구조를 그대로 사용한다. 예측하고 싶은 class 를 feature map 에 있는 각각의 원소값으로 미분해서 gradient 를 구한다. 한 feature map 의 모든 gradient 의 평균을 가중치로 정해 feature map 에 곱해주고,이 feature map 들을 합친 이미지에서 이 클래스로 예측하게 된 요인이 된 부분이 집중적으로 밝게 나타난다.



따라서 위 데이터의 CNN 결과를 토대로 CAM 을 진행하면, 일본어 히라가나 문자 10 개의 클래스 데이터이기 때문에 하얀 글씨 쪽에 집중적으로 밝게 나타날 것이고, 10 개의 클래스들 중에서 서로 상이하게 다른 모습을 보이는 부분들이 집중적으로 밝게 나타날 것이다.

3. SHAP 을 적용할 수 있는 간단히 예제를 만들어 보세요 (수업 시간 자료를 참고해서)

SHAP 은 게임이론에 기반해서 관측치마다 설명변수 각각이 사용됐을 때 대비 사용되지 않았을 때를 비교해 변수의 중요도를 결정하는 방법이다.

예를 들어서, 서울 성북구 안암동 지역의 집값을 예측한다고 할 때, 설명변수 3 개가 있다고 한다.

X1 : 해당 주택이 지어진 해

X2 : 평수

X3 : 동네의 치안율

이때 SHAP 을 적용하면 3 개 설명변수 중 주택가격에 있어서 가장 중요한 변수를 정할 수 있다. 각 변수의 shapley value 를 구해서 가장 큰 값을 가지는 변수가 가장 중요도가 높은 변수이다.

X1 변수의 shapley value 는 X1 변수를 사용했을 때의 예측값에 사용하지 않은 경우의 예측값을 뺀 수를 가중합 처리한 값이다. 이 shapley value 를 관측치별로 구할 수 있고, 이를 통해 가장 중요한 변수를 찾을 수 있다.

	X1 사용	X2 사용	X3 사용	예측값
Case1	X	X	X	20
2	O	X	X	23
3	X	O	X	24
4	X	X	O	25
5	O	O	X	24
6	O	X	O	26
7	X	O	O	28
8	O	O	O	30

X1 의 shapley value=(1/3)(23-20)+(1/6)(24-24)+(1/6)(26-25)+(1/3)(30-28)=1.833

X2, X3 의 shapley value 도 이와 마찬가지로 구할 수 있다.

X2 의 shapley value=(1/3)(24-20)+(1/6)(24-23)+(1/6)(28-25)+(1/3)(30-26)=3.333

X3 의 shapley value=(1/3)(25-20)+(1/6)(26-23)+(1/6)(28-24)+(1/3)(30-24)=4.833

X3 의 shapley value 가 가장 크므로 동네의 치안율이 집값에 큰 기여를 했다고 할 수 있다. 설명변수 중 가장 중요도가 낮은 변수는 해당 주택이 지어진 해이다. 하지만 SHAP 은 인과관계로 해석하기 어렵기 때문에 전후관계는 아니다.

4. 예측애널리틱스 수업에서 더 다루어 주었으면 하는 내용 (방법론)을 기탄없이 알려주세요 (다음에 본 수업을 들을 여러분 후배들을 위한 것이니 성의껏 명시해 주세요)

이번 예측애널리틱스 수업에서는 전통적인 데이터분석/머신러닝 기법과 최근 각광받고 있는 딥러닝 기법을 폭넓게 배울 수 있어서 매우 유익한 수업이었습니다. 보편적으로 많이 사용되는 방법론인 의사결정나무나 랜덤포레스트, 그리고 부스팅 등의 방법론 등까지 배울 수 있으면 더욱 좋을 것 같습니다.