Convolution Neural Network

(using cifar-10, fashion-mnist dataset)

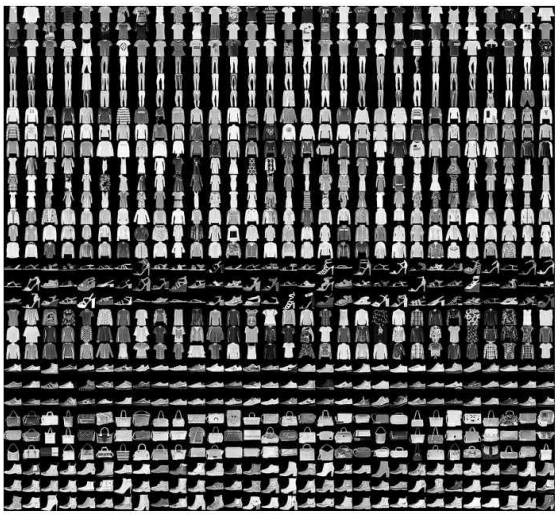
V2016120 김태형

Convolution Neural Network란?

-Deep Neural Network와 다르게 Convolution Layer가 추가, 즉, conv layer를 이용하여 각 이미지들의 feature를 추출하여 이 feature들이 각각의 weight들을 학습하는 Network이다.

Convolution Neural Network
(using Fashion-MNIST data)

Fashion-MNIST Data ?
- **Image size = 28 x 28**
- **Data Size [training set images = 60,000] , [test set images = 10,000]**
- **Data Label [0 ; T-shirt/top]**
  [1 ; Trouser]
  [2 ; Pullover]
  [3 ; Dress]
  [4 ; Coat]
  [5 ; Sandal]
  [6 ; Shirt]
  [7 ; Sneaker]
  [8 ; Bag]
  [9 ; Ankle boot]



Source : https://github.com/zalandoresearch/fashion-mnist

Processing

1. **Load Data**
2. **Define Model**
3. **Compile Model**
4. **Fit Model**
5. **Evaluate Model**
6. **Tie It All Together**

1~6번 순서로 진행하며, (2,3,4)번을 바꿔가면서 최적의 성능을 내는 CNN를 완성할 것이다.

Define Model

Conv2D(3x3)_32_relu_BN

|

MaxPooling2D(2x2)

|

Dropout(0.2)

|

Flatten

|

Dense(128) _relu_BN

|

Dense(10) _ softmax

|

Output

Conv2D(3x3)_32_relu_BN

|

Conv2D(3x3)_32_relu_BN

|

MaxPooling2D(2x2)

|

Dropout(0.25)

|

Conv2D(3x3)_64_relu_BN

|

Conv2D(3x3)_64_relu_BN

|

MaxPooling2D(2x2)

|

Dropout(0.25)

|

Flatten

|

Dense(512) _ relu_BN

|

Dropout(0.5)

|

Dense(10) _ softmax

|

Output

[두 가지 모델 비교]

Compile Model

Loss Function : Cross-entropy

Optimization: SGD

Learning Rate : 0.001

Loss Function : Cross-entropy

Optimization: Adam

Learning Rate : 0.001

Fit Model

Epoch = 200

Batch_size = 30

Epoch = 12

Batch_size = 100

Evaluate Model

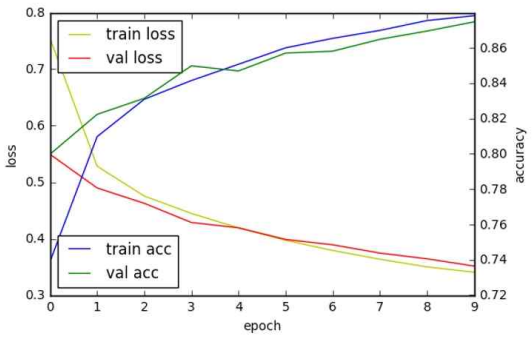
Accuracy : 85.71%

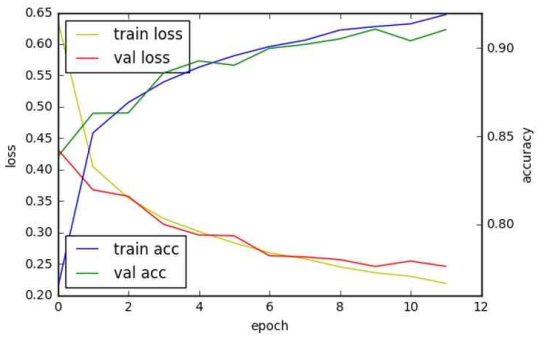
Val_Accuracy : 85.71%

Accuracy : 91.89%

Val_Accuracy : 91.04%

Visualization





Convolution Neural Network
(using CIFAR-10 data)

CIFAR-10 Data ?

- **Image size = 32 x 32 x 3**
- **Data Size [training set images = 50,000] , [test set images = 10,000]**
- **Data Label [0 ; Airplane]**
  [1 ; Automobile]
  [2 ; Bird]
  [3 ; Cat]
  [4 ; Deer]
  [5 ; Dog]
  [6 ; Frog]
  [7 ; Horse]
  [8 ; Ship]
  [9 ; Truck]

https://www.cs.toronto.edu/~kriz/cifar.html

Define Model

Conv2D(3x3)_32_relu_BN

|

MaxPooling2D(2x2)

|

Dropout(0.2)

|

Flatten

|

Dense(128) _relu_BN

|

Dense(10) _ softmax

|

Output

Conv2D(3x3)_32_relu_BN

|

Conv2D(3x3)_32_relu_BN

|

MaxPooling2D(2x2)

|

Dropout(0.25)

|

Conv2D(3x3)_64_relu_BN

|

Conv2D(3x3)_64_relu_BN

|

MaxPooling2D(2x2)

|

Dropout(0.25)

|

Flatten

|

Dense(512) _ relu_BN

|

Dropout(0.5)

|

Dense(10) _ softmax

|

Output

[두 가지 모델 비교]

Compile Model

Loss Function : Cross-entropy

Optimization: SGD

Learning Rate : 0.001

Loss Function : Cross-entropy

Optimization: Adam

Learning Rate : 0.001

Fit Model

Epoch = 200

Batch_size = 30

Epoch = 12

Batch_size = 100

Evaluate Model

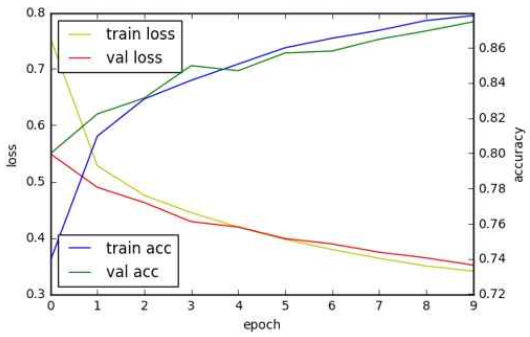
Accuracy : 85.71%

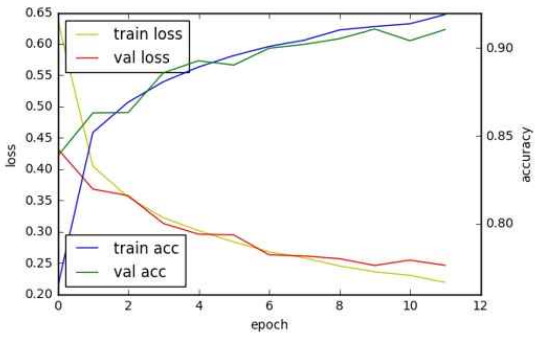
Val_Accuracy : 85.71%

Accuracy : 91.89%

Val_Accuracy : 91.04%

Visualization





Batch Normalization (ICML 2015)

Deep Learning에서 골치 아픈 문제 중 하나는 vanishing/exploding gradient 문제

Layer 수가 적은 경우는 문제가 심각하지 않지만, layer 수가 많아지면 누적될수록 문제가 심각해 진다

이러한 문제를 해결하기 위해서 Dropout, Regularization 등 방법이 있지만 여전히 Layer 수가 많아지면 Training이 잘 안되는 경향이 있어서,

나온 것이 BN(Batch Normalization)이다.
BN에서는 Training하는 과정 자체를 안정화 및 속도를 향상시키고 싶어 했고,
이들은 불 안정화가 일어나는 이유가 "Internal Covariance Shift"라고 주장

Internal Covariance Shift ? 이전 layer의 parameter 변화로 인해 현재 layer의 분포가 바뀌는 현상

해결방법으로는? Batch normalization, Whitening

Whitening : input분포의 평균을 0, 표준편차 1인 input으로 normalization시키는 방법

단순하게 whitening을 통해 평균과 분산을 조정하는 것보다는 좀 더 나은 방법이 필요하며, 그것이 바로 BN(batch normalization)
BN은 평균과 분산을 조정하는 과정이 별도의 process로 있는 것이 아니고,
Training하면서 같이 조절이 된다는 점이 whitening과 차이점

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Training, Test 차이점

Training ->mini-batch 마다 값을 구함

Test ->구한 값의 평균을 사용