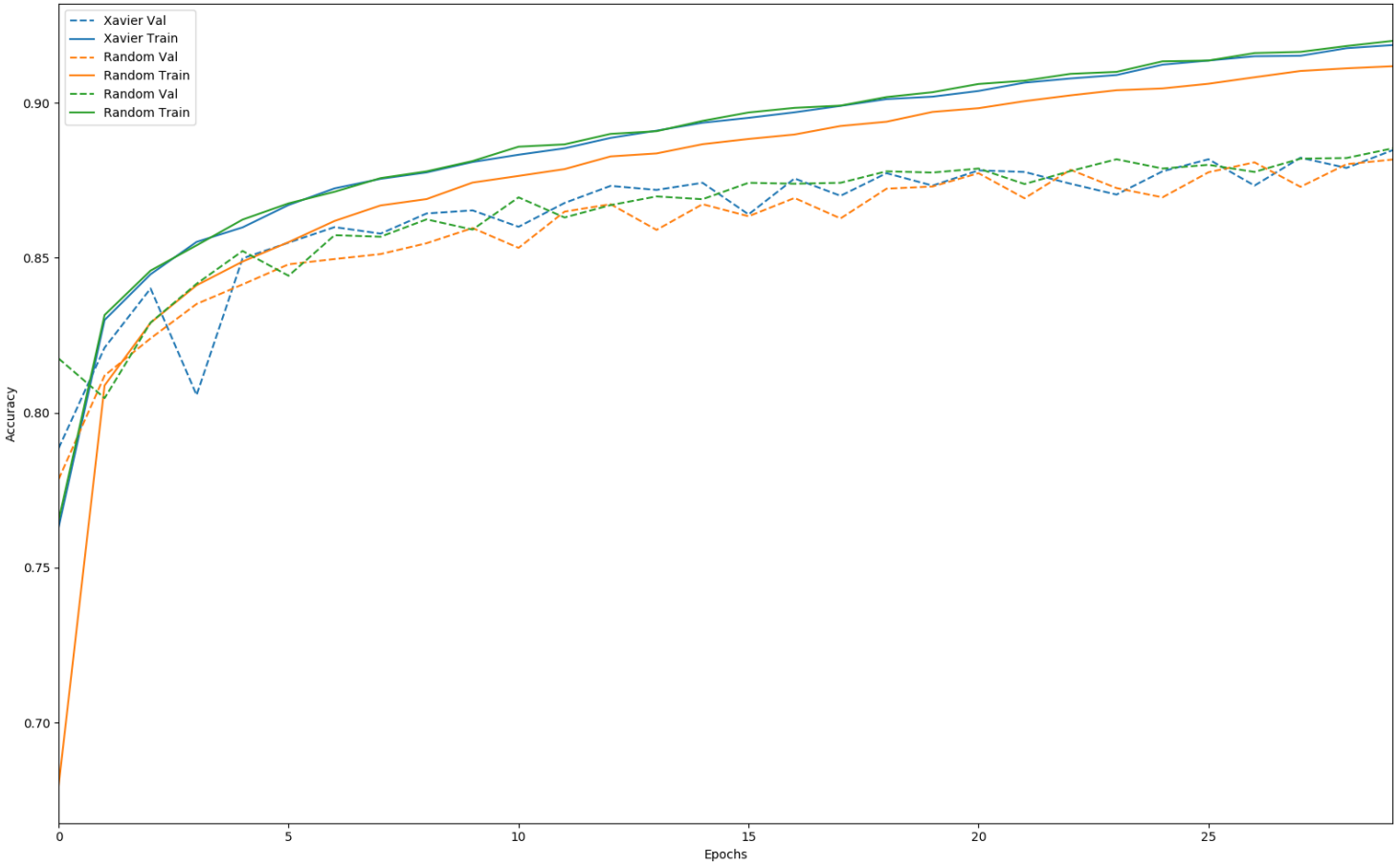
Dropout & Batch Normalization

명지대학교
MYONGJI UNIVERSITY

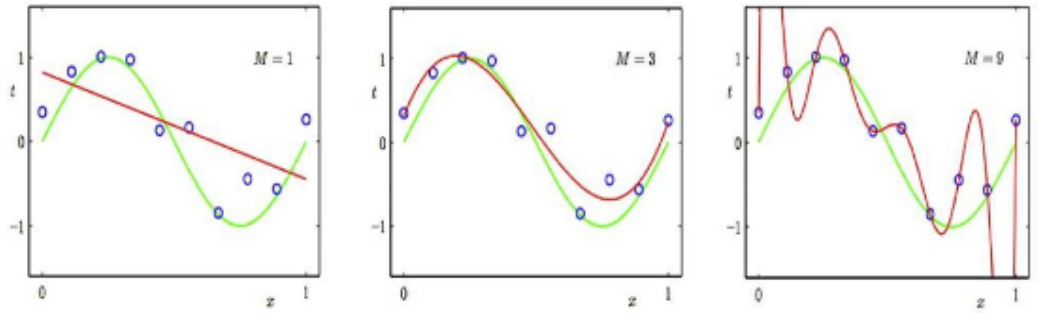
Weight 초기화


Xavier Val
Xavier Train
Random Val
Random Train
Random Val
Random Train
Accuracy
0.90
0.85
0.80
0.75
0.70
0
5
10
15
20
25
Epochs

과적합(Overfitting)

Under-fitting | Good | Overfitting

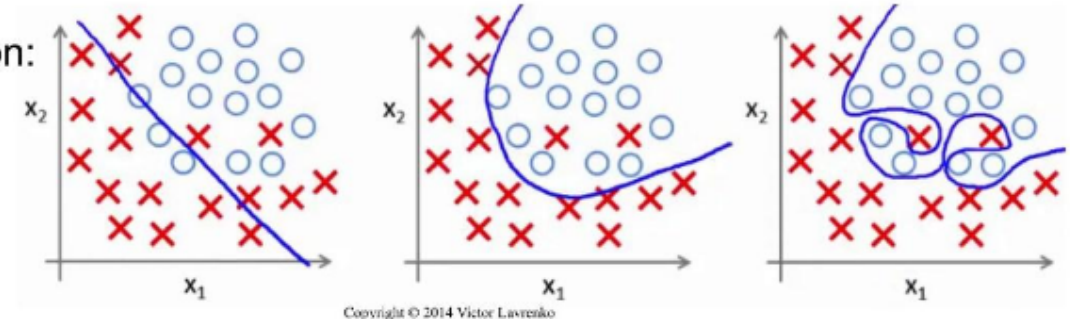
Regression:



predictor too inflexible: cannot capture pattern

predictor too flexible: fits noise in the data

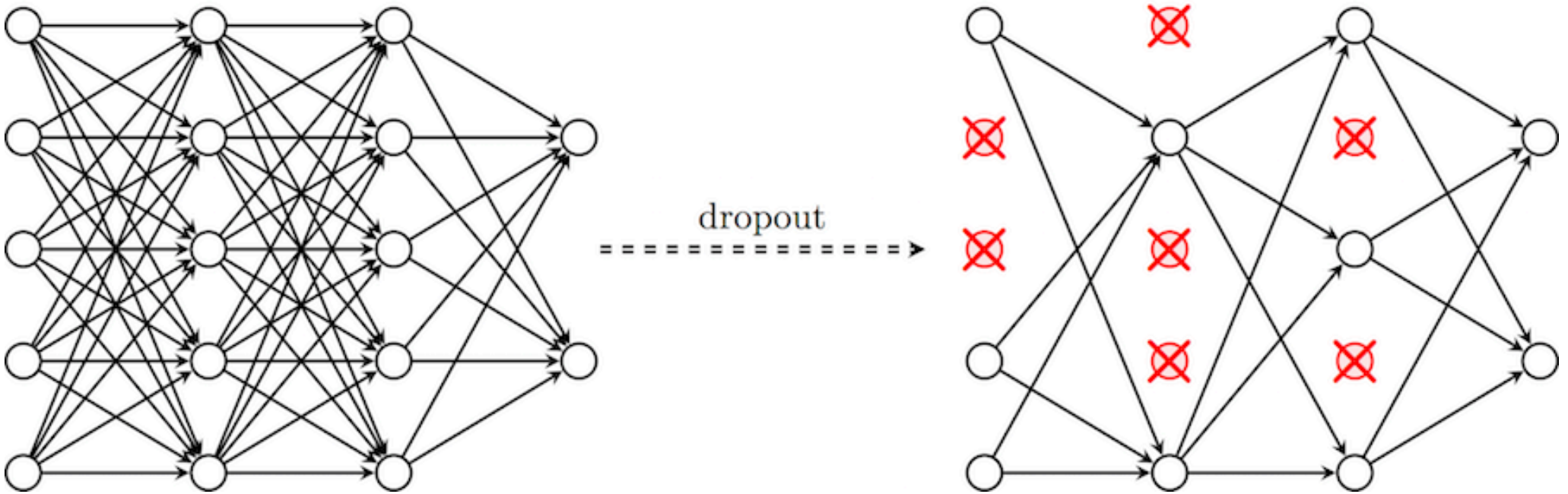
Classification:



Training ⇩ Test ⇩

Training ⇧ Test ⇩

Dropout



https://sungjk.github.io/2018/03/11/convnet-2.html

Original Model

```
def makemodel(X_train, y_train, X_valid, y_valid, weight_init):
    model = keras.models.Sequential()

    model.add(keras.layers.Flatten(input_shape=[28, 28]))
    model.add(dense(300, weight_init, activation="relu"))
    model.add(dense(100, weight_init, activation="relu"))
    model.add(dense(10,weight_init,  activation="softmax"))



    model.summary()
    return model
```

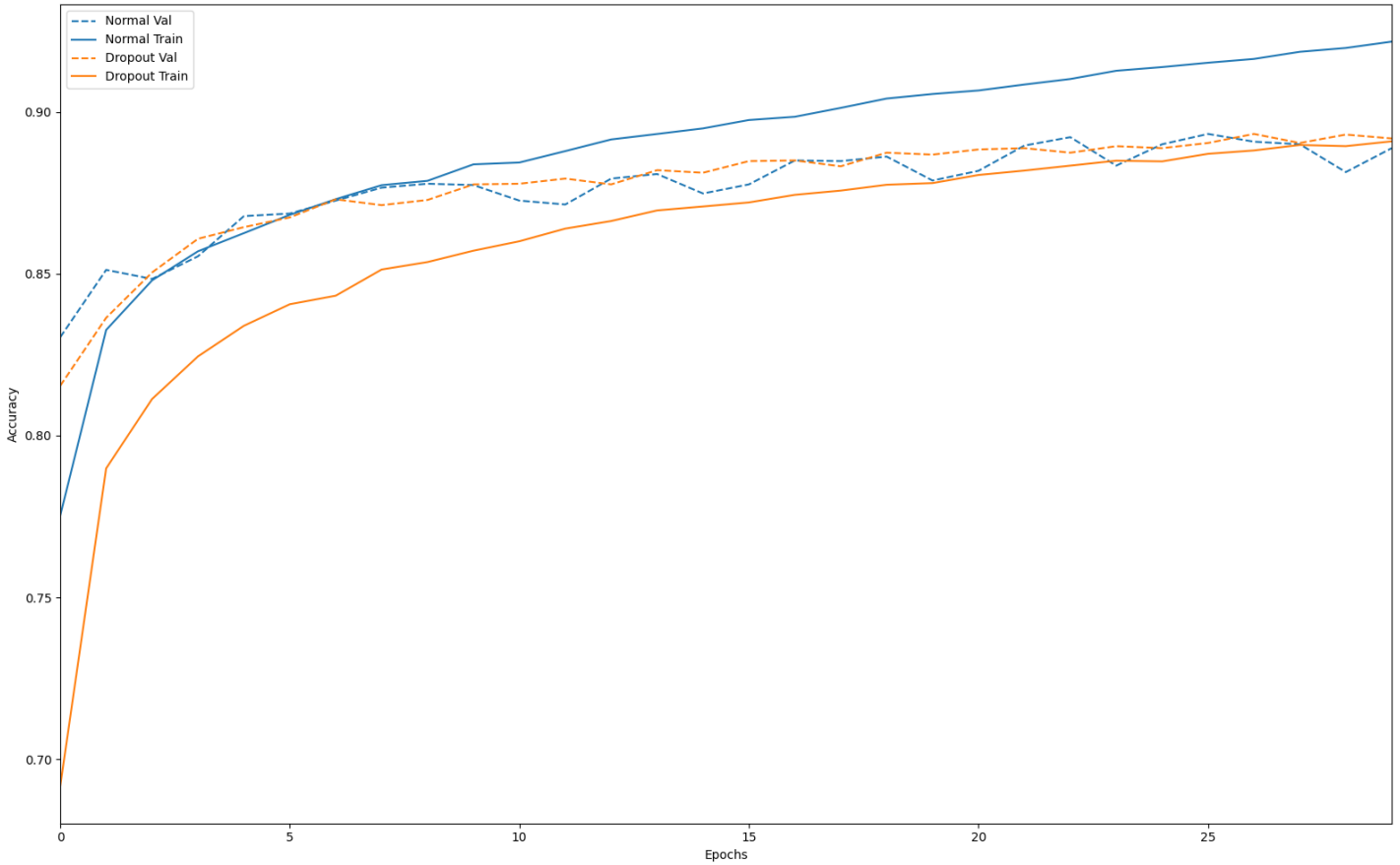
Dropout model

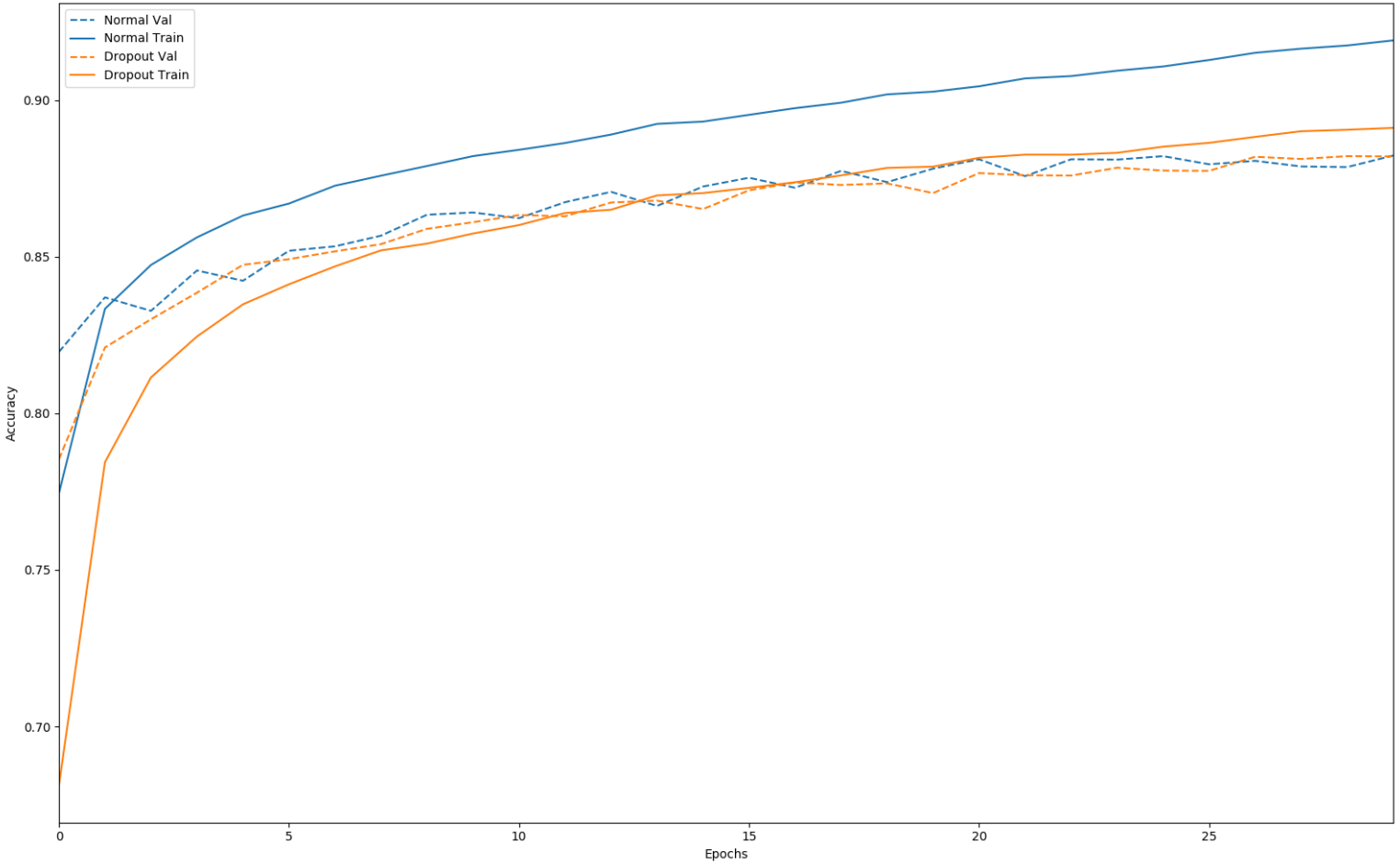
```
def makemodeldrop(X_train, y_train, X_valid, y_valid, weight_init):
    model = keras.models.Sequential()

    model.add(keras.layers.Flatten(input_shape=[28, 28]))
    model.add(dense(300, weight_init, activation="relu"))
    model.add(keras.layers.Dropout(0.3))
    model.add(dense(100, weight_init, activation="relu"))
    model.add(keras.layers.Dropout(0.3))
    model.add(dense(10,weight_init,  activation="softmax"))


    model.summary()
    return model
```

Dropout model

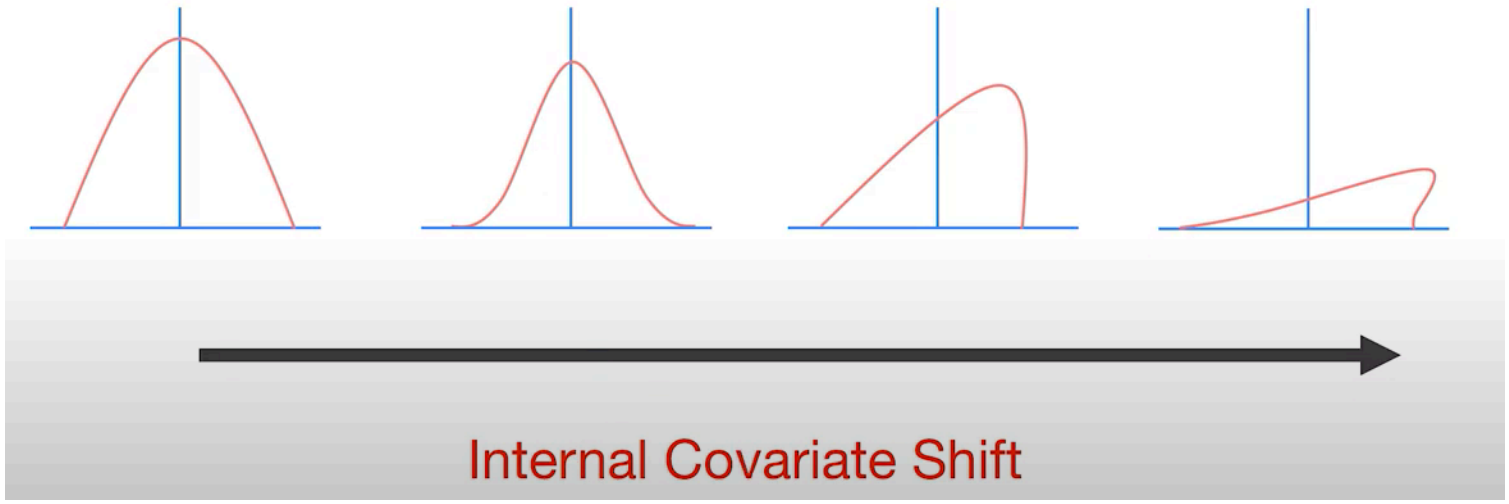




명지대학교
MYONGJI UNIVERSITY

Batch Normalization

- Batch Normalization은 각각의 스칼라 Feature들을 독립적으로 정규화
- 각각의 Feature들의 Mean 및 Variance를 **0**과 **1**로 정규화



Batch Normalization

```
def makemodelbatch(X_train, y_train, X_valid, y_valid, weight_init):
    model = keras.models.Sequential()

    model.add(keras.layers.Flatten(input_shape=[28, 28]))
    model.add(dense(300, weight_init, activation="relu"))
    model.add(keras.layers.BatchNormalization())
    model.add(dense(100, weight_init, activation="relu"))
    model.add(keras.layers.BatchNormalization())
    model.add(dense(10,weight_init,  activation="softmax"))


    model.summary()
    return model
```

Batch Normalization