ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

NATIONAL CENTRE FOR
SCIENTIFIC RESEARCH "DEMOKRITOS"

Deep Learning
Image Classification In Fashion Data

Karlis Vassilis
Toliopoulou Christianna

The Problem

- Classification of images in fashion data
- Why?
  - A potential business use case is an application which tries to find similar clothes, sorted by price and location
- How?
  - Logistic Regression (baseline). Just a simple experiment
  - MLPs
  - CNNs
  - A Combination of MLPs and CNNs

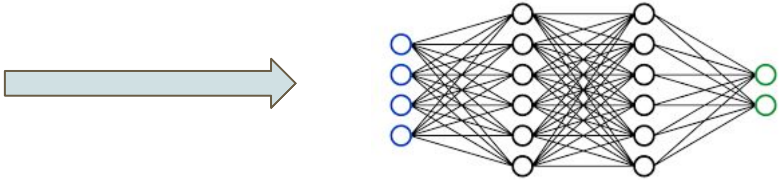
The Data

- Public Data
  - Source: Zalando research / Available in Tensorflow
  - Easy to experience with different methodologies
- Data Info
  - 70.000 greyscale fashion images
  - 10 classes
- Pre-processing workflow

Split dataset

Scale pixels [0,1]

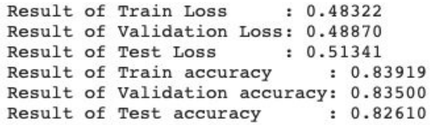
Convert to categorical



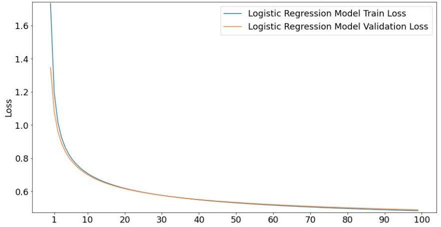
Models 1/4

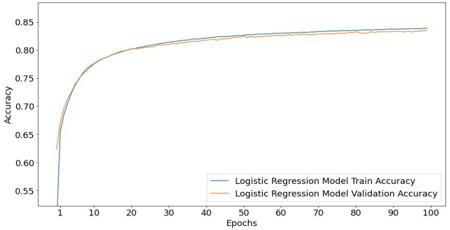
Logistic Regression

*SGD Optimizer (learning rate = 0.1)
*0 Hidden Layers
*Hidden Activation Function: Relu
*Output activation : Softmax



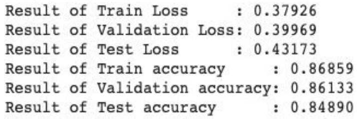
```
Result of Train Loss      : 0.48322
Result of Validation Loss: 0.48870
Result of Test Loss       : 0.51341
Result of Train accuracy       : 0.83919
Result of Validation accuracy: 0.83500
Result of Test accuracy         : 0.82610
```



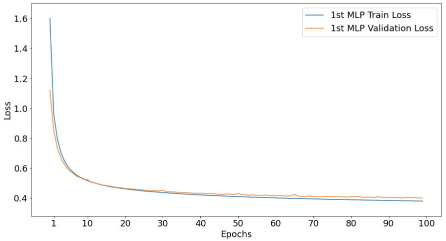


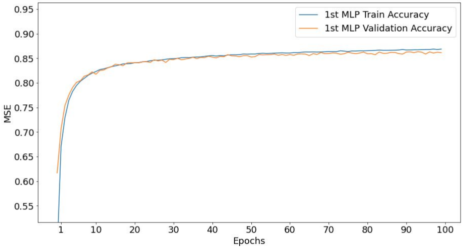
1st MLP

* activation function: relu
* hidden layers: 1
* nodes of hidden layer: 10
* nodes of output layer: 10
* Epochs: 100
* Loss Function: categorical cross-entropy
* optimizer :SGD



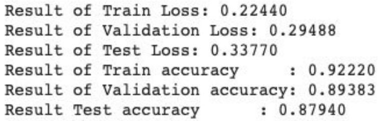
```
Result of Train Loss      : 0.37926
Result of Validation Loss: 0.39969
Result of Test Loss       : 0.43173
Result of Train accuracy      : 0.86859
Result of Validation accuracy: 0.86133
Result of Test accuracy       : 0.84890
```



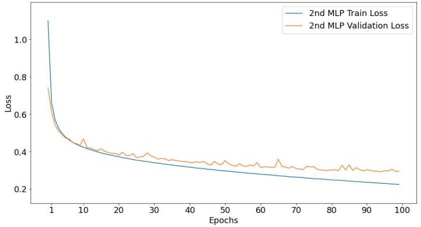


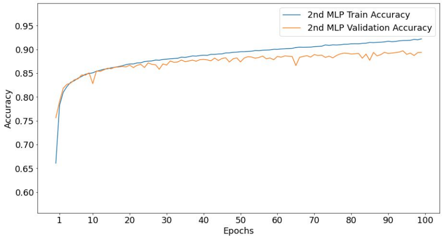
2nd MLP

* hidden layers (first layer: 256 nodes, second layer: 128 nodes)
* 1 output layer(10 nodes)
* Activation function: relu
* Loss function -> categorical cross-entropy
* optimizer :SGD



```
Result of Train Loss: 0.22440
Result of Validation Loss: 0.29488
Result of Test Loss: 0.33770
Result of Train accuracy     : 0.92220
Result of Validation accuracy: 0.89383
Result Test accuracy        : 0.87940
```

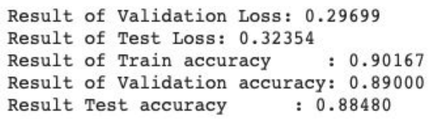




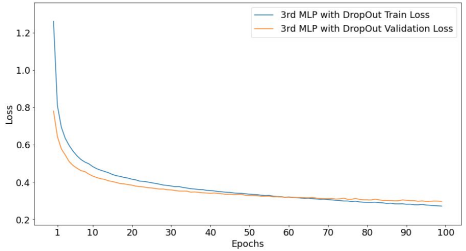
Models 2/4

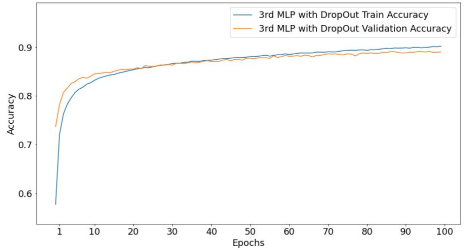
3rd MLP with dropout

*hidden layers 2 (first layer: 256 nodes,   second layer: 128 nodes)
* drop out : 0.2
*Activation function: relu
*Loss function -> categorical cross-entropy
*optimizer :SGD



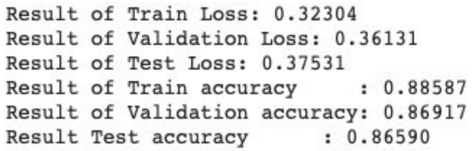
```
Result of Validation Loss: 0.29699
Result of Test Loss: 0.32354
Result of Train accuracy     : 0.90167
Result of Validation accuracy: 0.89000
Result Test accuracy       : 0.88480
```



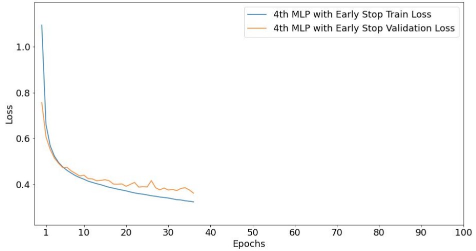


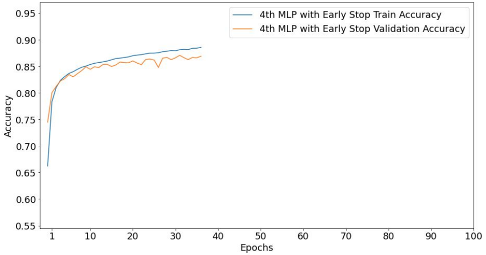
4th MLP with early stopping

*hidden layers 2 (256,128)
*Activation function:relu
*Loss function:categorical cross entropy
*optimizer SGD
*Running Epochs: 37



```
Result of Train Loss: 0.32304
Result of Validation Loss: 0.36131
Result of Test Loss: 0.37531
Result of Train accuracy     : 0.88587
Result of Validation accuracy: 0.86917
Result Test accuracy       : 0.86590
```

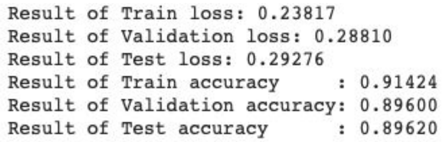




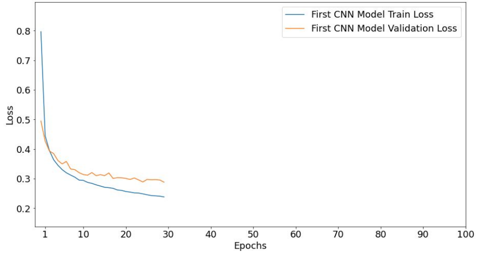
Models 3/4

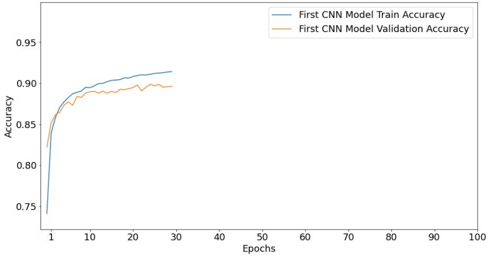
1st CNN

* Convolutional Layers: 2
* Drop out : 0.2
* Activation function: relu
* Loss function -> categorical cross-entropy
* Optimizer : Adam
* Kernel Size: 2
* Filter Dimension: 8



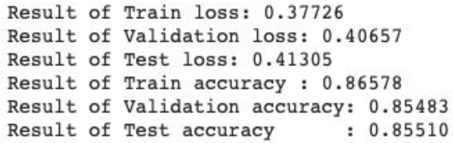
```
Result of Train loss: 0.23817
Result of Validation loss: 0.28810
Result of Test loss: 0.29276
Result of Train accuracy     : 0.91424
Result of Validation accuracy: 0.89600
Result of Test accuracy      : 0.89620
```



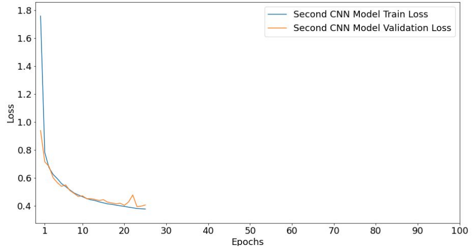


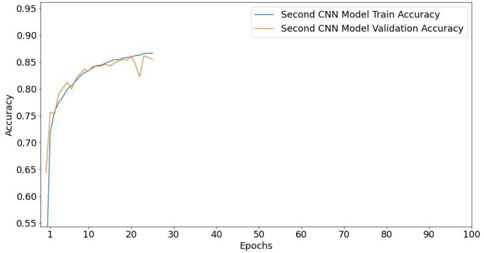
2nd CNN

* Convolutional Layers: 2
* Drop out : 0.2
* Activation function: relu
* Loss function -> categorical cross-entropy
* Optimizer : SGD
* Kernel Size: 2
* Filter Dimension: 8



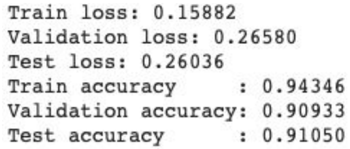
```
Result of Train loss: 0.37726
Result of Validation loss: 0.40657
Result of Test loss: 0.41305
Result of Train accuracy : 0.86578
Result of Validation accuracy: 0.85483
Result of Test accuracy      : 0.85510
```



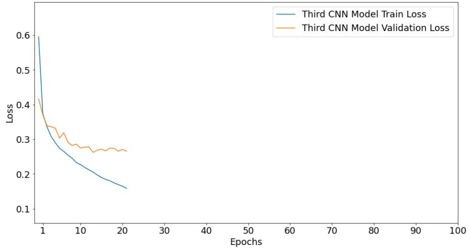


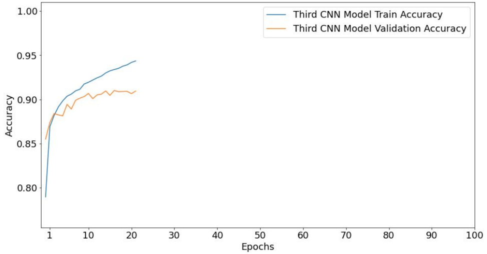
3rd CNN

* Convolutional Layers: 2
* Drop out : 0.2
* Activation function: relu
* Loss function -> categorical cross-entropy
* Optimizer : SGD
* Kernel Size: 2
* Filter Dimension: 32



```
Train loss: 0.15882
Validation loss: 0.26580
Test loss: 0.26036
Train accuracy      : 0.94346
Validation accuracy: 0.90933
Test accuracy       : 0.91050
```

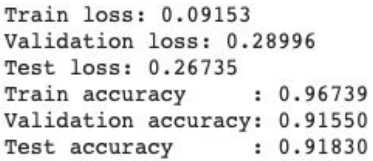




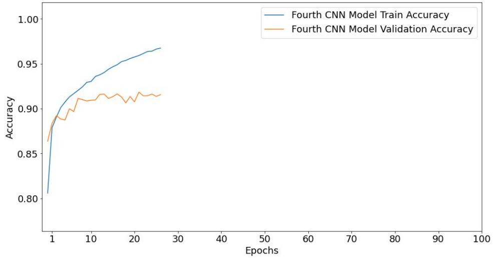
Models 4/4

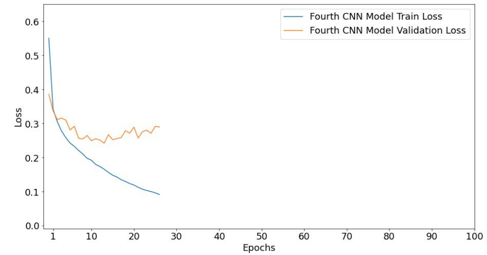
4th CNN

*Convolutional Layers: 2
*Drop out : 0.2
*Activation function: relu
*Loss function -> categorical cross-entropy
*Optimizer : Adam
*Kernel Size: 3
*Filter Dimension: 32



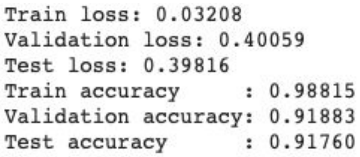
```
Train loss: 0.09153
Validation loss: 0.28996
Test loss: 0.26735
Train accuracy      : 0.96739
Validation accuracy: 0.91550
Test accuracy       : 0.91830
```



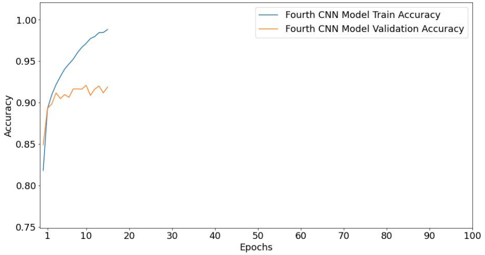


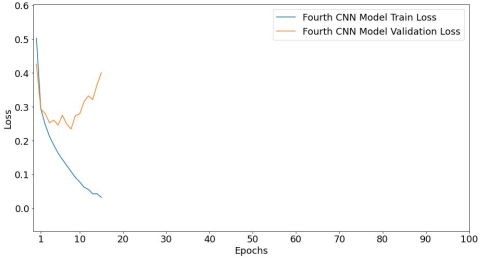
5th CNN

*Convolutional Layers: 4
*Drop out : 0.2
*Activation function: relu
*Loss function -> categorical cross-entropy
*Optimizer : Adam
*Kernel Size: 3
*Filter Dimension: 64



```
Train loss: 0.03208
Validation loss: 0.40059
Test loss: 0.39816
Train accuracy      : 0.98815
Validation accuracy: 0.91883
Test accuracy       : 0.91760
```





Results ½ - MLPs

Model	Hidden Layer	Nodes of hidden layer	Early stopping	Dropout	Validation Loss	Validation Accuracy	Test Accuracy
1st MLP	1	10	no	no	42%	85%	84%
2nd MLP	2	256/128	no	no	33%	88%	88%
3rd MLP	2	256/128	no	yes/0.2	30%	89%	88%
4th MLP	2	256/128	yes	no	34%	88%	87%

COMMON CHARACTERISTICS ACROSS

- 100 epochs
- Cross entropy loss function
- Input relu
- Output softmax

Results 2/2 - CNN

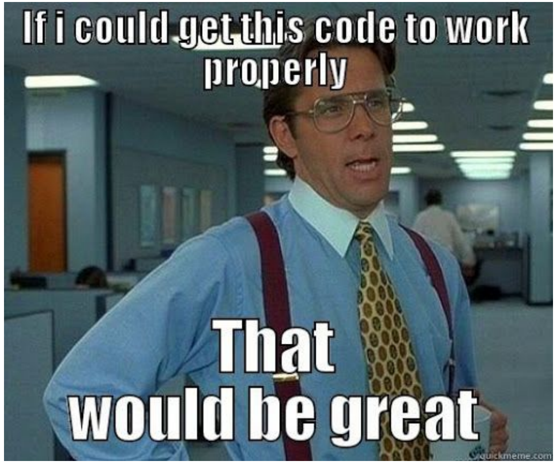
Model	Filter Dimension	Conv.Layers	Kernel filter	Optimizer	Validation Loss	Validation Accuracy	Test Accuracy	Dropout
1st CNN	8	2	2	Adam	29%	89%	89%	yes/ 0.2 - 30 epochs
2nd CNN	8	2	2	SGD	40%	85%	85%	yes/ 0.2 - 26 epochs
3rd CNN	32	2	2	Adam	26%	90%	91%	yes/0.2 - 22 epochs
4th CNN	32	2	3	Adam	28%	91%	91%	yes/0.2 - 27 epochs
5th CNN	64	4	3	Adam	39%	91%	91%	yes/0.2 - 16 epochs

COMMON CHARACTERISTICS ACROSS

- Cross entropy loss function
- Convolutional activation relu
- Output activation softmax

Challenges/ Learnings

- Computational problems
- Coding experience
- Small differences in results -> experiencing with multiple parameters
- Theoretical background



Future Work - Conclusion

- A combination of CNN with MLP
- Experiments using momentum in various optimizers
- Early stopping in validation loss