

Assignment 3

Name	ID
Yomna Taher Abdallah	20190624
Adel Abdelmonem Arfa	20190280

In this Assignment we use CNN with minst dataset so we will try many models to get the best one of these data and we attached the code with this file.

For first we load the data, and then we change the shape of output to vector with 1 in the index of the correct number and zeros for others.

Because we will use CNN, we will include 1 dim for filter operation and separate the data to train and test, so our data shape will be:

- Train (60000, 28, 28, 1)
- Test (10000, 28, 28, 1)

We know that CNN need to be tested to get the best model, so let's try our models and choose the best one.

➤ Let's try our **model_1**: (Epochs changing)

- Final accuracy: 96.46% and first 5 epoch (74.46%, 92.06%, 93.69%, 94.51%, 95.09%)
- Total parameters: 175,946
- epochs=10
- batch_size=32
- learning_rate=0.01
- The average time to train: 10~11s
- The average time to test :2s
- loss = 0.1162
- In this model we use 2 CNN layers and 2 FC layers with activation function (relu) in all of them and (softmax) for last layer
- SGD optimizer

➤ Let's try our **model_2**:

- Final accuracy: **96.86%** and first 5 epoch (73.99%, 91.98%, 93.59%, 94.43%, 94.99%)
- Total parameters : **175,946**
- epochs=**12**
- batch_size=**32**
- learning_rate=**0.01**
- The average time to train: **10~11s**
- The average time to test :**2s**
- loss = **0.1053**
- In this model we use **2 CNN layers and 2 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
- **SGD** optimizer

➤ Let's try our **model_3**:

- Final accuracy: **97.44%** and first 5 epoch (70.12%, 92.12%, 93.75%, 94.66%, 95.12%)
- Total parameters : **175,946**
- epochs=**15**
- batch_size=**32**
- learning_rate=**0.01**
- The average time to train: **10~11s**
- The average time to test :**2s**
- loss = **0.0813**
- In this model we use **2 CNN layers and 2 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
- **SGD** optimizer

As we see in first three models (1,2,3) with changing the number of epochs it effects on the accuracy of the model and on model loss, so we will use **15 epochs** in coming models.

-
- Let's try our **model_4**: (learning_rate)
- Final accuracy: **98.15%** and first 5 epoch (87.36%, 94.82%, 96.69%, 97.50%, 98.02%)
 - Total parameters : **175,946**
 - epochs=**15**
 - batch_size=**32**
 - learning_rate=**0.05**
 - The average time to train: **10~11s**
 - The average time to test :**2s**
 - loss = **0.0800**
 - In this model we use **2 CNN layers and 2 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
-

- Let's try our **model_5**:
- Final accuracy: **98.14%** and first 5 epoch (88.20%, 95.36%, 97.13%, 98.44%, 98.77%)
 - Total parameters : **175,946**
 - epochs=**15**
 - batch_size=**32**
 - learning_rate= **0.08**
 - The average time to train: **10~11s**
 - The average time to test :**2s**
 - loss = **0.0787**
 - In this model we use **2 CNN layers and 2 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
-

- **Let's try our model_6**:
- Final accuracy: **98.17%** and first 5 epoch (82.42%, 93.52%, 94.85%, 95.79%, 96.50%)
 - Total parameters: **175,946**
 - epochs=**15**
 - batch_size=**32**
 - learning_rate= **0.03**
 - The average time to train:**10~11s**

- The average time to test :2s
- loss = 0.0691
- In this model we use 2 CNN layers and 2 FC layers with activation function (relu) in all of them and (softmax) for last layer
- SGD optimizer

As we see in three models (4,5,6) with changing the learning rate it effects on the accuracy of the model and on model loss, so we will use 0.03 in coming models.

➤ Let's try our model_7: (Adding removing layers)

- Final accuracy: 97.42% and first 5 epoch (97.33%, 97.51%, 97.61%, 97.78%, 97.87%)
 - Total param s: 43,530
 - epochs=15
 - batch_size=32
 - learning_rate= 0.03
 - The average time to train: 8s
 - The average time to test: 1s
 - loss = 0.0906
 - In this model we use 1 CNN layers and 1 FC layers with activation function (relu) in all of them and (softmax) for last layer
 - SGD optimizer
-

➤ Let's try our model_8:

- Final accuracy: 97.34% and first 5 epoch (82.70%, 92.52%, 94.25%, 95.21%, 95.84%)
- Total parameters : 173,946
- epochs=15
- batch_size=32
- learning_rate= 0.03
- The average time to train: 7~8s
- The average time to test :1s
- loss = 0.0871

- In this model we use **1 CNN layers and 2 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
-

➤ Let's try our **model_9**:

- Final accuracy: **98.13%** and first 5 epoch (85.61%, 93.83%, 95.14%, 96.03%, 96.72%)
 - Total parameters : **175,578**
 - epochs=**15**
 - batch_size=**32**
 - learning_rate= **0.03**
 - The average time to train: **10~11s**
 - The average time to test : **2s**
 - loss = **0.0582**
 - In this model we use **2 CNN layers and 1 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
-

➤ Let's try our **model_10**:

- Final accuracy: **98.45%** and first 5 epoch (84.02%, 96.14 %, 97.33%, 97.96%, 98.28%)
 - Total parameters: **160,346**
 - epochs=**15**
 - batch_size=**32**
 - learning_rate= **0.03**
 - The average time to train: **13~14s**
 - The average time to test :**2s**
 - loss = **0.0534**
 - In this model we use **3 CNN layers and 2 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
-

➤ Let's try our **model_11**:

- Final accuracy: **98.49%** and first 5 epoch (82.07%, 96.03%, 97.16%, 97.72%, 98.09%)
- Total parameters: **160,714**
- epochs=**15**
- batch_size=**32**
- learning_rate= **0.03**
- The average time to train: **14s**
- The average time to test :**2s**
- loss = **0.0572**
- In this model we use **3 CNN layers and 3 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
- **SGD** optimizer

As we see in five models (7,8,9,10,11) with changing the **numbers of layers** it not effects on the accuracy of the model, model loss only it also effects on the number of parameters which refers to the amount of memory we use, so we will use **3 CNN layers and 2 FC layers** in coming models.

➤ Let's try our **model_12**: (batch size)

- Final accuracy: **98.79%** and first 5 epoch (87.87%, 97.00 %, 97.87%, 98.32%, 98.68%)
 - Total parameters: **160,346**
 - epochs=**15**
 - batch_size=**20**
 - learning_rate= **0.03**
 - The average time to train: **18~19s**
 - The average time to test :**2s**
 - loss = **0.0620**
 - In this model we use **3 CNN layers and 2 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
-

➤ Let's try our **model_13**:

- Final accuracy: **97.44%** and first 5 epoch (75.51%, 94.57%, 96.53%, 97.25%, 97.70%)
- Total parameters: **160,346**
- epochs=**15**
- batch_size=**64**
- learning_rate= **0.03**
- The average time to train: **8~9s**
- The average time to test :**2s**
- loss = **0.0824**
- In this model we use **3 CNN layers and 2 FC layers** with activation function (**relu**) in all of them and (**softmax**) for last layer
- **SGD** optimizer

As we see in tow models (12,13) with changing the **batch size** it not effects on the accuracy of the model, model loss only it also effects on the average time of training but we chose to focus on the accuracy, so we will use **20 batch size** in coming models.

➤ Let's try our **model_14**: (**activation function**)

- Final accuracy: **11.35%** and first 5 epoch (10.59%, 10.53%, 10.74%, 10.90%, 10.87%)
 - Total parameters: **160,346**
 - epochs=**15**
 - batch_size=**20**
 - learning_rate= **0.03**
 - The average time to train: **17~18s**
 - The average time to test :**2s**
 - loss = **2.3012**
 - In this model we use **3 CNN layers and 2 FC layers** with activation function (**sigmoid**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
-

➤ Let's try our **model_15**:

- Final accuracy: **98.86%** and first 5 epoch (90.60%, 97.20%, 98.21%, 98.66%, 98.95%)
 - Total parameters: **160,346**
 - epochs=**15**
 - batch_size=**20**
 - learning_rate= **0.03**
 - The average time to train: **17~18s**
 - The average time to test :**2s**
 - loss = **0.0379**
 - In this model we use **3 CNN layers and 2 FC layers** with activation function (**tanh**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
-

➤ Let's try our **model_16**:

- Final accuracy: **98.85%** and first 5 epoch (87.32%, 95.65%, 97%, 97.72%, 98.21%)
- Total parameters: **160,346**
- epochs=**15**
- batch_size=**20**
- learning_rate= **0.03**
- The average time to train: **18~19s**
- The average time to test :**2s**
- loss = **0.0381**
- In this model we use **3 CNN layers and 2 FC layers** with activation function (**softsign**) in all of them and (**softmax**) for last layer
- **SGD** optimizer

As we see in three models (14,15,16) with changing the **activation function** it effects on the accuracy of the model, model loss, so we will use **tanh function** in coming models.

- Let's try our **model_17**: (optimizers)
- Final accuracy: **98.51%** and first 5 epoch (95.56%, 98.19%, 98.57%, 98.82%, 98.99%)
 - Total parameters: **160,346**
 - epochs=**15**
 - batch_size=**20**
 - The average time to train: **19~20s**
 - The average time to test :**2s**
 - loss = **0.0553**
 - In this model we use **3 CNN layers and 2 FC layers** with activation function (**tanh**) in all of them and (**softmax**) for last layer
 - **Adam** optimizer
-

- Let's try our **model_18**:
- Final accuracy: **98.76%** and first 5 epoch (95.74%, 98.32%, 98.83%, 99.15%, 99.30%)
 - Total parameters: **160,346**
 - epochs=**15**
 - batch_size=**20**
 - learning_rate= **0.03**
 - The average time to train: **23s**
 - The average time to test :**2s**
 - loss = **0.0796**
 - In this model we use **3 CNN layers and 2 FC layers** with activation function (**tanh**) in all of them and (**softmax**) for last layer
 - **RMSprop** optimizer

As we see in tow models (17,18) with changing the **optimizer** it not effects on the accuracy of the model, model loss only it also effects on the average time of training, so we will use **SGD** optimizer (**model 15**) in coming models.

➤ Let's try our **model_19**: (Dropout location)

- Final accuracy: **98.66%** and first 5 epoch (87.92%, 95.29%, 96.61%, 97.12%, 97.64%)
 - Total parameters: **160,346**
 - epochs=**15**
 - batch_size=**20**
 - learning_rate= **0.03**
 - The average time to train: **19s**
 - The average time to test :**2s**
 - loss = **0.0444**
 - In this model we use **3 CNN layers and 2 FC layers** with activation function (**tanh**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
 - We put the drop out after **the first FC layer**
 - Dropout rate: **50%**
-

➤ Let's try our **model_20**:

- Final accuracy: **98.83%** and first 5 epoch (88.26%, 96.09%, 97.34%, 97.91%, 98.30%)
- Total parameters: **160,346**
- epochs=**15**
- batch_size=**20**
- learning_rate= **0.03**
- The average time to train: **19s**
- The average time to test :**2s**
- loss = **0.0433**
- In this model we use **3 CNN layers and 2 FC layers** with activation function (**tanh**) in all of them and (**softmax**) for last layer
- **SGD** optimizer
- We put the drop out after **the second FC layer**
- Dropout rate: **50%**

As we see in tow models (19,20) with changing the **Dropout location** it effects on the accuracy of the model, model loss, so we will use **after second layer** in coming models.

➤ Let's try our **model_21: (Dropout Rate)**

- Final accuracy: **98.98%** and first 5 epoch (98.10%, 98.60%, 98.76%, 99.02%, 99.15%)
 - Total parameters: **160,346**
 - epochs=**15**
 - batch_size=**20**
 - learning_rate= **0.03**
 - The average time to train: **19s**
 - The average time to test :**2s**
 - loss = **0.0349**
 - In this model we use **3 CNN layers and 2 FC layers** with activation function (**tanh**) in all of them and (**softmax**) for last layer
 - **SGD** optimizer
 - We put the drop out after **the second FC layer**
 - Dropout rate: **25%**
-

➤ Let's try our **model_22:**

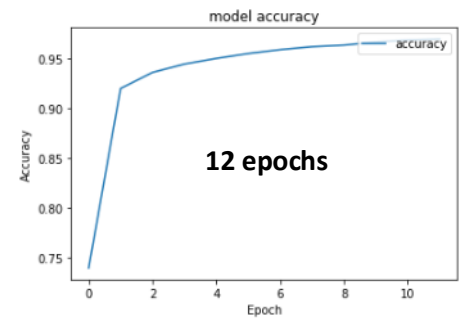
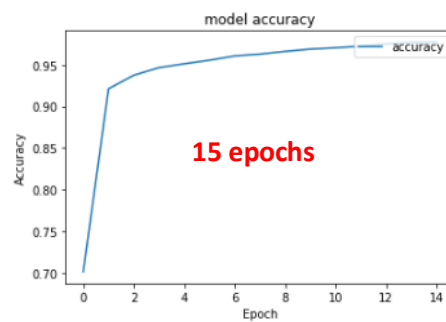
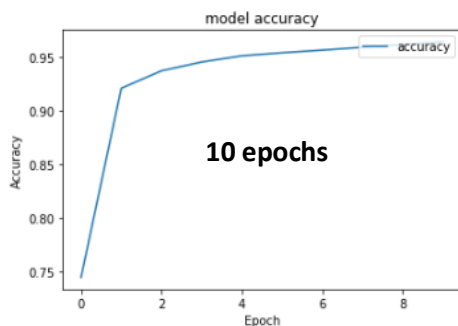
- Final accuracy: **98.59%** and first 5 epoch (84.04%, 93.62%, 95.54%, 96.11%, 96.54%)
- Total parameters: **160,346**
- epochs=**15**
- batch_size=**20**
- learning_rate= **0.03**
- The average time to train: **19s**
- The average time to test :**2s**
- loss =**0.0648**
- In this model we use **3 CNN layers and 2 FC layers** with activation function (**tanh**) in all of them and (**softmax**) for last layer
- **SGD** optimizer
- We put the drop out after **the second FC layer**
- Dropout rate: **70%**

As we see in two models (21,22) with changing the **Dropout rate** it effects on the accuracy of the model, model loss, so we will use **25%**.

let's summarize all of this in some points:

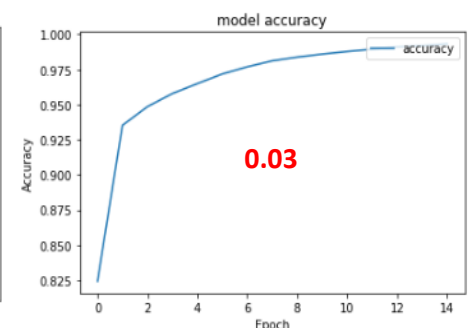
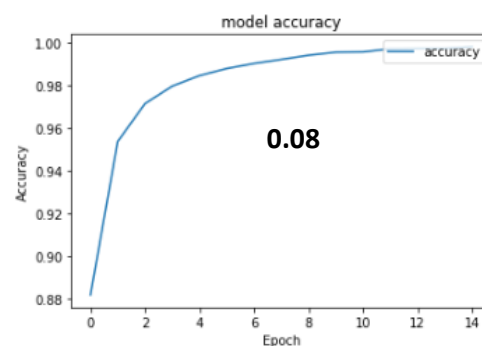
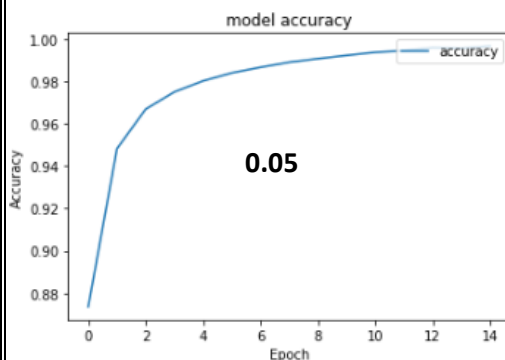
- in the first three models (1,2,3), we build our models with freezing all things that effect on the model except the **epochs**, we use three different numbers of them (10,12,15) and this make difference in two things (**accuracy, loss**).

So, we chose the best one that give the best result which was (15) with accuracy 97.44 and loss 0.0823



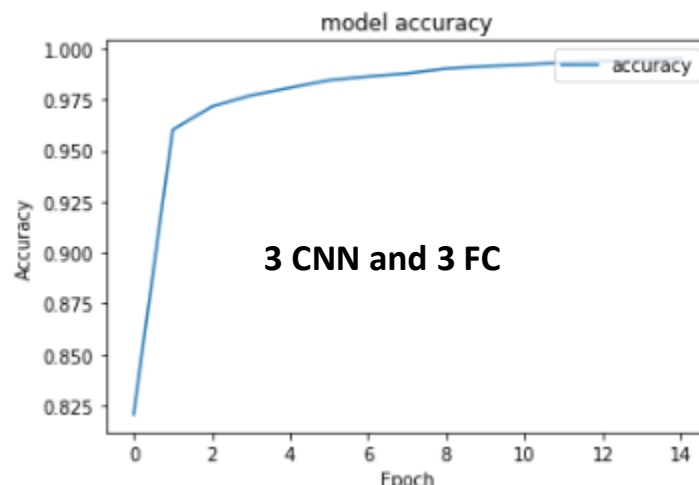
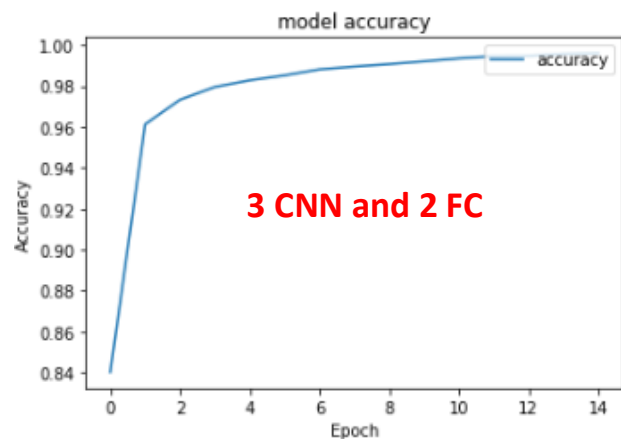
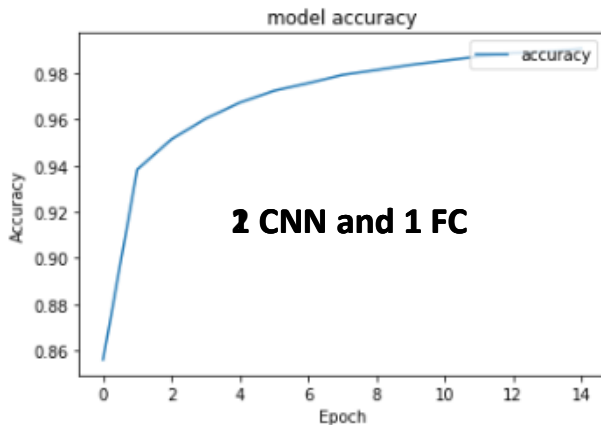
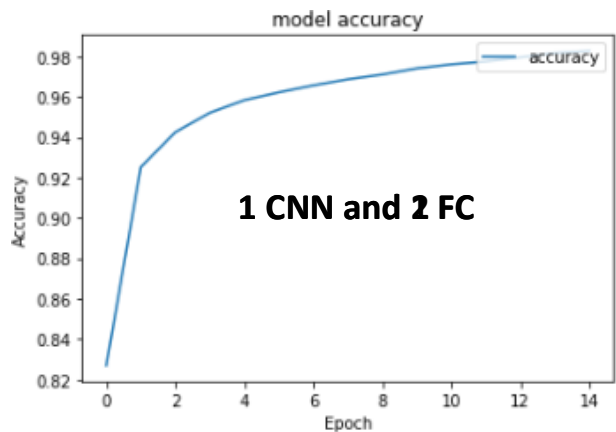
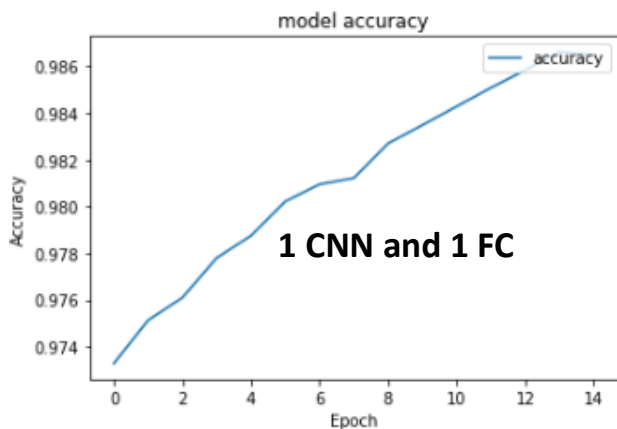
- In second three models (4,5,6) we frizzed all things that effects on the model except the **learning rate**, we used three learning rate (0.05,0.03,0.08) and this make difference in two things (**accuracy, loss**).

So, we chose the best one that give the best result which was (0.03) with accuracy 98.17 and loss 0.0691



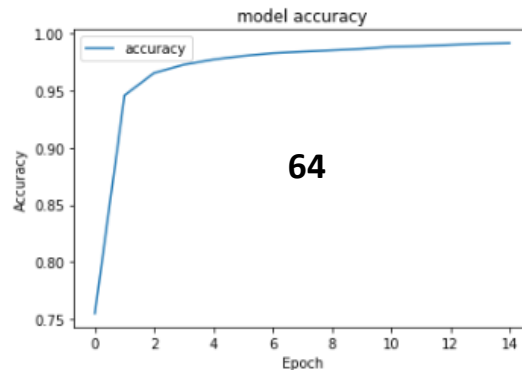
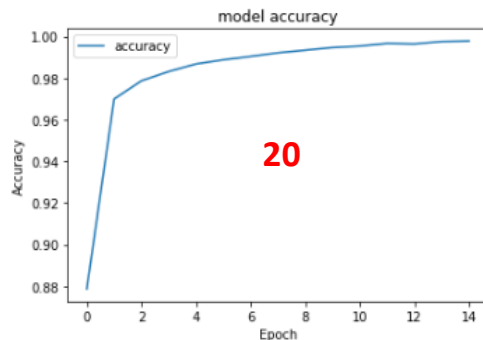
- in next five models (7,8,9,10,11) we frizzed all things that effects on the model except the **removing adding layers** , we used five cases (1 CNN and 1 FC \ 1 CNN and 2 FC \ 2 CNN and 1 FC \ 3 CNN and 2 FC \ 3 CNN and 3 FC) and this make difference in all result of the models (**accuracy, loss, time of training, number of parameters**).

So, we chose the best one that give the best result which was (3 CNN and 2 FC) with accuracy **98.45**, loss **0.0534**, the average time to train **13~14s**



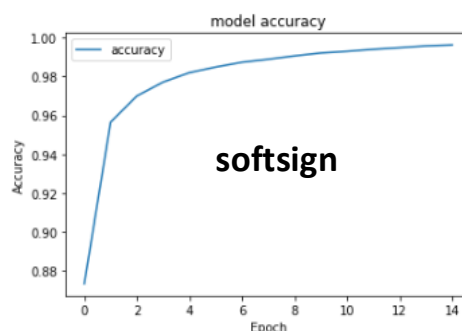
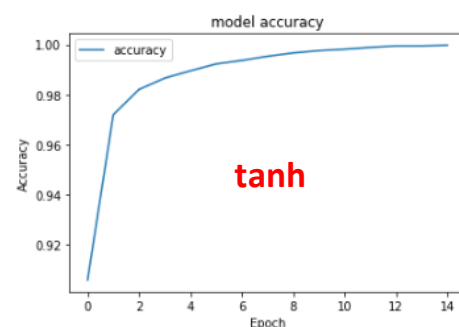
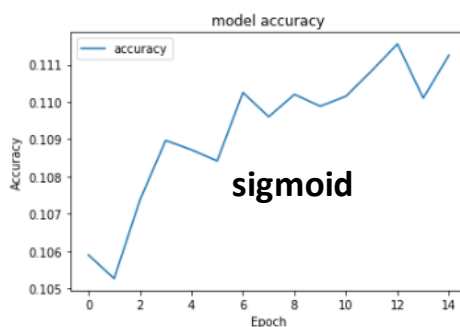
- In the next two models (12,13) , we build our models with freezing all things that effect on the model except the **batch size**, we use three different numbers of them (20,32,64) and this make difference in all result of the models (**accuracy, loss, time of training**) .

So, we chose the best one that give the best result which was (20) with accuracy 98.79 and loss 0.0620 and the average time to train 18~19s



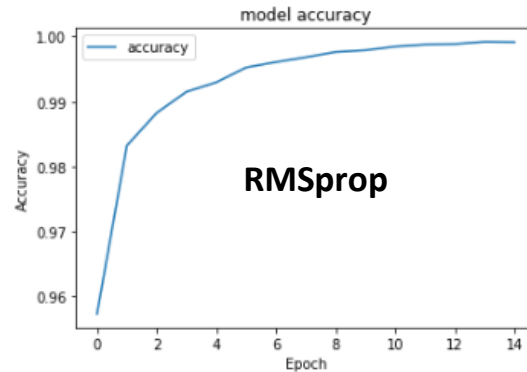
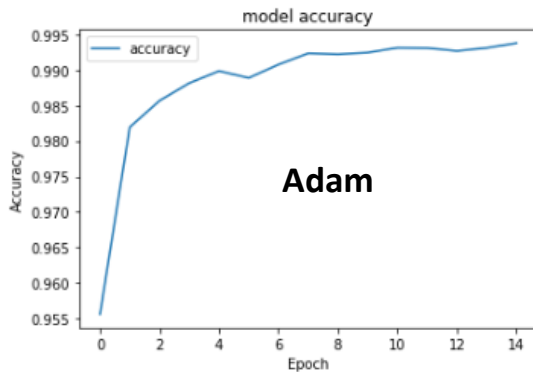
- In the next three models (14,15,16) , we build our models with freezing all things that effect on the model except the **activation function**, we use three different types of it (**sigmoid,tanh,softsign**) and this make difference in all result of the models (**accuracy, loss**).

So, we chose the best one that give the best result which was (tanh) with accuracy 98.86 and loss 0.0379



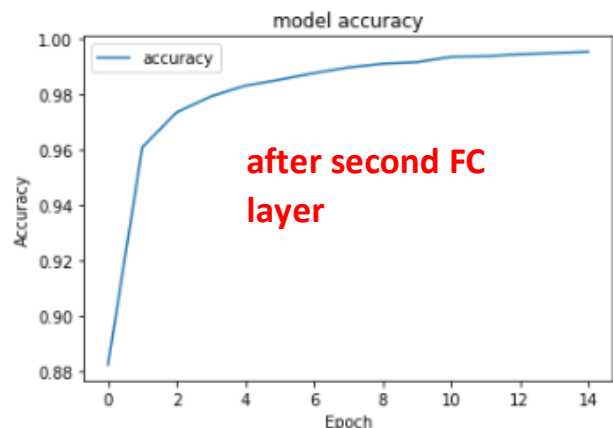
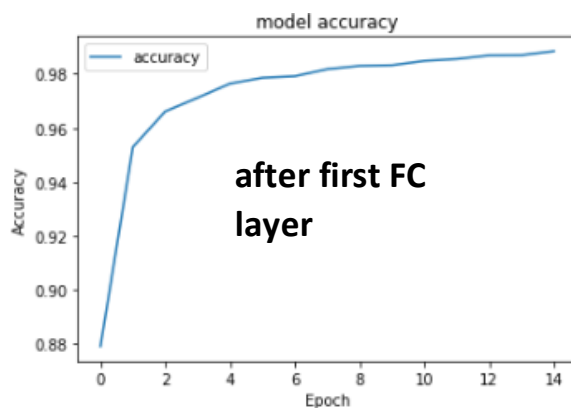
- in the next two models (17,18) , we build our models with freezing all things that effect on the model except the **optimizer**, we use two different types including the **SGD** of it (**Adam, RMSprop**) and this make difference in all result of the models (**accuracy, loss, time of training**).

So, we chose the best one that give the best result, which was **SGD** with accuracy **98.86**, loss **0.0379** and the average time to train **17~18s**



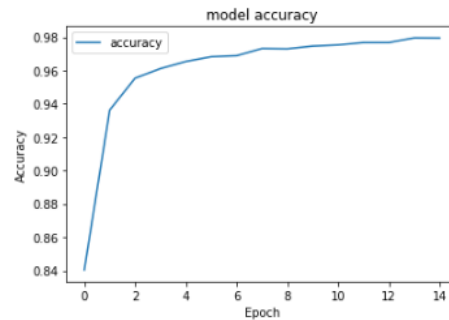
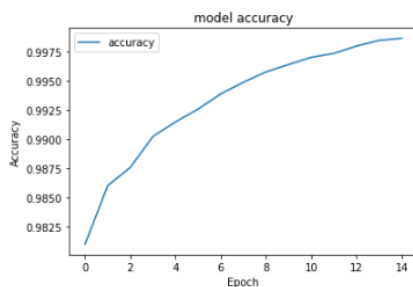
- In the next two models (19,20) , we build our models with freezing all things that effect on the model except **the dropout layer location**, we use two different locations of it (**after first FC layer , after second FC layer**) and this make difference in all result of the models (**accuracy, loss**)

So, we chose the best one that give the best result, which was **after second FC layer** with accuracy **98.83**,and loss **0.0433**



- In the next two models (21,22), we build our models with freezing all things that effect on the model except the **dropout rate**, we use two rates including 0.5 (0,25, 0.7) and this make difference in all result of the models (accuracy, loss)

So, we chose the best one that give the best result, which was 0.25 with accuracy 98.98, and loss 0.0349



➤ Now let's see our best mode:

Best	Model
Final accuracy	98.98%
Total parameters	160,346
epochs	15
batch_size	20
he average time to train	19s
The average time to test	2s
loss	0.0349
Number of CNN & FC	3 CNN & 2 FC
activation function	tanh
optimizer	SGD
Learning rate	0.03
Dropout location	After the second FC layer
Dropout rate	25%

