**Επίλυση προβλήματος ταξινόμησης με χρήση Multilayer-Perceptron δικτύου (MLP).**

*Τριφηνόπουλος Χρήστος*

Μέρος πρώτο: Μελέτη του δικτύου για διάφορους συνδυασμούς παραμέτρων.

Parameters:

Hidden Layer activation: Relu

Output layer activation: Softmax

Objective function: Accuracy

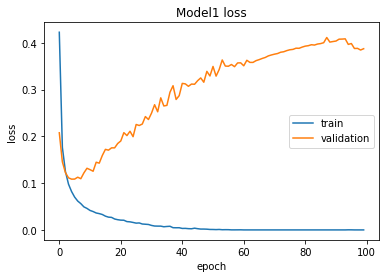
Optimization: Categorical Cross entropy

Minibatch size: 256

Training epochs: 100

1. RMSProp Optimizer (learning rate=0.001, rho=0.01)

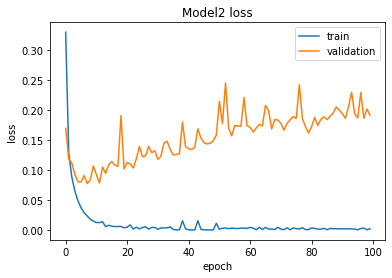
Graphical user interface

Description automatically generated with medium confidence

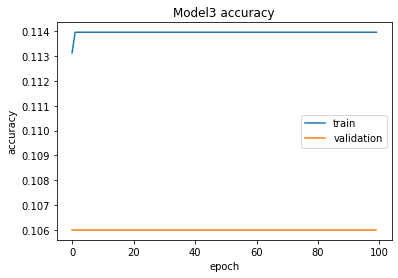
**Evaluation on training set: loss: 0.3121 - accuracy: 0.9787**

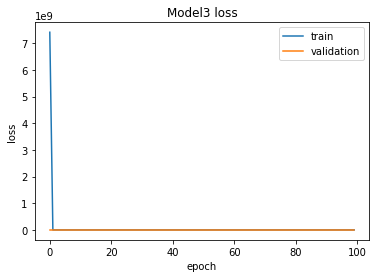
1. RMSProp Optimizer (learning rate =0.001, rho=0.99)

Text, whiteboard

Description automatically generated

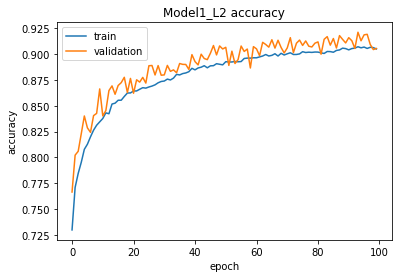
**Evaluation on training set: loss: 0.1716 - accuracy: 0.9829**

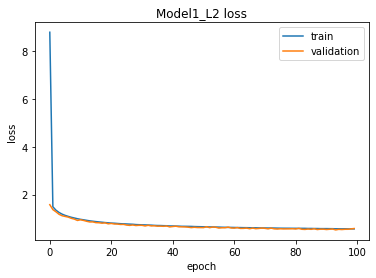
1. SGD optimizer (learning rate =0.01, weights initialized randomly from normal distribution with mean equal to 10)



**Evaluation on training set: loss: 2.3010 - accuracy: 0.1135**

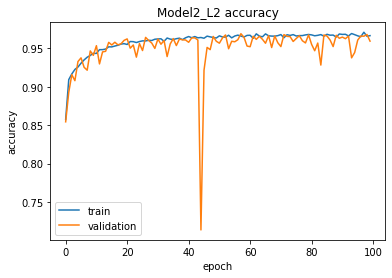
1. RMSProp Optimizer (learning rate=0.001, rho=0.01, L2 Regularization with a=0.1)

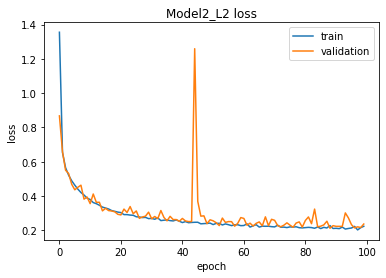




**Evaluation on training set: loss:0.4775 - accuracy: 0.9221**

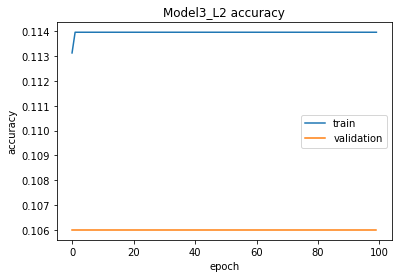
1. RMSProp Optimizer (learning rate =0.001, rho=0.99, L2 Regularization with a=0.01)

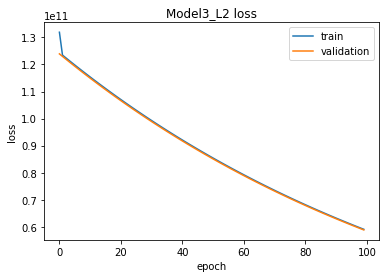




**Evaluation on training set: loss:0.1957 - accuracy: 0.9697**

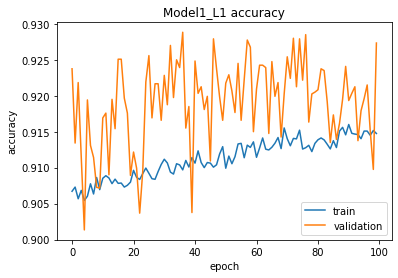
1. SGD optimizer (learning rate =0.01, weights initialized randomly from normal distribution with mean equal to 10, L2 Regularization with a=0.001)

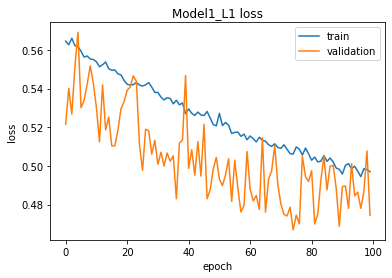




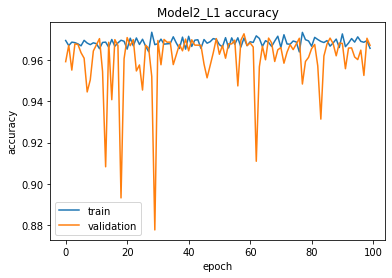
**Evaluation on training set: loss: 27746269184.0000 - accuracy: 0.1135**

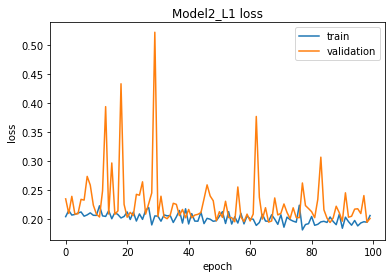
1. RMSProp Optimizer (learning rate=0.001, rho=0.01, L1 Regularization with a=0.01, plus dropout layer with propability=0.3)



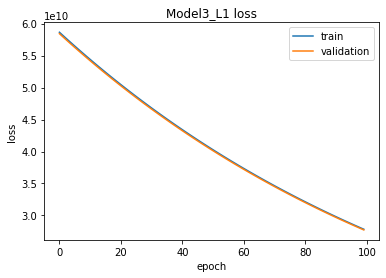
**Evaluation on training set: loss: 98.9698 - accuracy: 0.1200**

1. RMSProp Optimizer (learning rate =0.001, rho=0.99, L1 Regularization with a=0.01, plus dropout layer with propability=0.3)





**Evaluation on training set: loss: 99.0333 - accuracy: 0.0958**

1. SGD optimizer (learning rate =0.01, weights initialized randomly from normal distribution with mean equal to 10, L1 Regularization with a=0.01, plus dropout layer with propability=0.3)

**Evaluation on training set: loss:4052765.0000 - accuracy: 0.0958**

**Σημείωση: Τα αποτελέσματα στις περιπτώσεις 3,6,9 δείχνουν ότι το δίκτυο αδυναμεί να εκπαιδευτεί (underfitting) καθώς το accuracy παραμένει σταθερό με το χρόνο. Αυτό πιθανόν συμβαίνει επειδή αρχικοποιούμε τα βάρη με πολύ μεγάλες τιμές σε σχέση με το learning rate. (Το επιβεβαίωσα αρχικοποιώντας τα βάρη με mean=0 και το testing accuracy για την περίπτωση 3&6 ανέβηκε στο 95% )**

**To overfitting στα δεδομένα εκπαίδευσης εμφανίζεται ως μια διαφορά στο accuracy του training με του validation set. Όταν η καμπύλη του training accuracy βρίσκεται σταθερά πάνω από αυτήν του validation accuracy όπως γίνεται στην περίπτωση 1 και 2 το μοντέλο κάνει overfit. Στο learning curve βλέπουμε αντίστοιχα το validation loss να μειώνεται μέχρι κάποιο epoch και μετά αρχίζει να αυξάνεται ενώ το training loss συνεχίζει να μειώνεται.**

**Από τα μοντέλα που δοκιμάσαμε την καλύτερη εικόνα παίρνουμε στις περιπτώσεις 4 και 5 όπου με την βοήθεια του L2 regularization έχει μειωθεί σημαντικά το overfitting. Αυτό όμως δεν συναπάγεται και με καλύτερο accuracy στο testing set καθώς weight penalty που εισάγει η κανονικοποίηση μειώνει αρκετά το accuracy (ειδικά στην περίπτωση 4).**

**Καθώς τα μοντέλα χωρίς κανονικοποίηση πετυχαίνουν καλύτερη ακρίβεια στο testing set, και μεταξύ των 4&5 αυτό με την μικρότερη παράμετρο α (περίπτωση 5) είναι το βέλτιστο μπορούμε να πούμε ότι η κανονικοποίηση δεν είναι καλή πρακτική για το συγκεκριμένο νευρωνικό.**

**Το dropout layer λειτουργεί παρόμοια με την κανονικοποίηση προσθέτοντας θόρυβο για να αποφευχθεί το overfitting. Βλέπουμε ότι και αυτό χειροτερεύει σημαντικά το μοντέλο μας. Για 2 dropout layers (ένα μετά από κάθε hidden layer), αντί για 1, το accuracy πέφτει περεταίρω.**

Μέρος δεύτερο: Fine tuning του δικτύου με χρήση keras\_tuner.

Οι υπερπαραμέτροι που ελέγχθηκαν είναι:

1. Αριθμός νευρώνων πρώτου κρυφού στρώματος(64,128)
2. Αριθμός νευρώνων δεύτερου κρυφού στρώματος(256,512)
3. Παράμετρος κανονικοποίησης(10^-1,10^-3,10^-6)
4. Ρυθμός εκμάθησης(10^-1,10^-2,10^-3)

Οι βέλτιστες τιμές που προέκυψαν είναι:

1)Optimal node amount for first hidden layer: 128

2)Optimal node amount for second hidden layer: 256

3)Optimal L2 parameter: 10^-6

4)Optimal learning rate: 10^-3

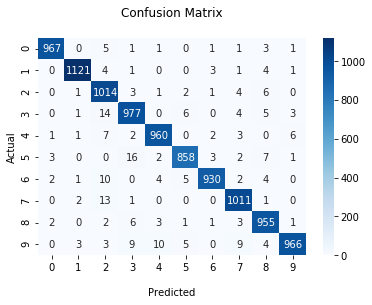
Ελέγχοντας το βέλτιστο μοντέλο στο testing set παίρνουμε:

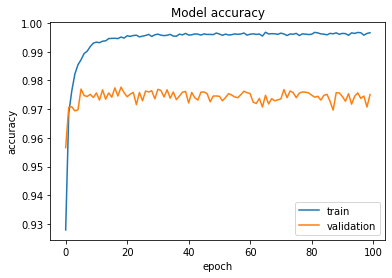
accuracy: 0.9758999943733215

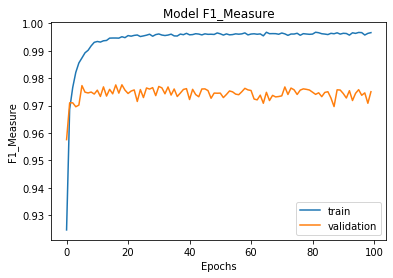
recall: 0.9759384989738464

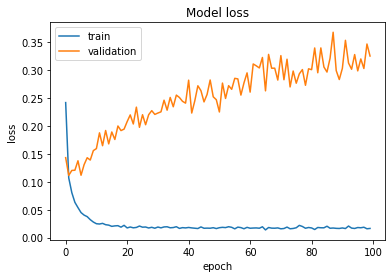
precision: 0.9760383367538452

F1\_Measure: 0.9759875535964966









Και σε αυτήν την περίπτωση έχουμε overfitting καθώς συμβαίνει το ίδιο με τις περιπτώσεις 1 και 2 του πρώτου μέρους αλλά το σφάλμα τόσο για το training όσο και για το validation παραμένει αρκετά μικρό. Αυτό σε συνδυασμό με το γεγονός ότι το accuracy στο testing set είναι αρκετά υψηλό μας δείχνει ότι παρά το overfitting, το μοντέλο είναι αρκετά καλό στο να “διαβάζει” το συγκεκριμένο dataset.

Όπως ήταν αναμενόμενο από τα αποτελέσματα του πρώτου μέρους, το βέλτιστο μοντέλο έχει τον ελάχιστο συντελεστή κανονικοποίησης των βαρών από αυτούς που δοκιμάσαμε.

Όσον αφορά το learning rate η πήρε την μικρότερη τιμή στο βέλτιστο μοντέλο κάτι το οποίο πιθανότατα συμβαίνει επειδή ο RMSPROP χρησιμοποιεί και ορμή οπότε έχει την δυνατότητα να κάνει μεγαλύτερα βήματα και να απεγκλωβιστεί από τοπικά ελάχιστα.